

Data mining para el descubrimiento de fenotipos en fetos con restricción de crecimiento a partir de datos pre y post natales

Nombre Estudiante: Sara Quesada Gil

Plan de Estudios del Estudiante: Máster de Ciencia de Datos (Data Science)

Área del trabajo final: Área 3

Nombre Consultor/a: Elisenda Bonet Carne y Xavier Paolo Burgos Artizzu

Nombre Profesor/a responsable de la asignatura: Elisenda Bonet Carne y Xavier Paolo Burgos Artizzu

Fecha Entrega: 24/06/2020

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2020 Sara Quesada Gil

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Sara Quesada Gil)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Descripción del trabajo</i>
Nombre del autor:	<i>Sara Quesada Gil</i>
Nombre del consultor/a:	<i>Elisenda Bonet Carne y Xavier Paolo Burgos Artizzu</i>
Nombre del PRA:	<i>Elisenda Bonet Carne y Xavier Paolo Burgos Artizzu</i>
Fecha de entrega (mm/aaaa):	03/2020
Titulación:	<i>Máster en Ciencia de Datos (Data Science)</i>
Área del Trabajo Final:	<i>Área 3: Data mining para el descubrimiento de fenotipos en fetos con restricción de crecimiento a partir de datos pre y post natales</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>data-mining, data-processing, modeling, statistics, deep-learning, machine-learning, FGR [1]</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i>	
<p>El objetivo de este proyecto es detectar patrones en los datos ofrecidos por BCNatal Fetal Medicine Research Center con el fin de averiguar si la relación entre estos es determinante o explicativa de la Restricción de Crecimiento Intra-Uterino (FGR) [1], y la prematuridad, las dos causas más significativas de morbilidad neonatal.</p> <p>La motivación para realizar el proyecto es personal, ya que opino que la sociedad no está suficientemente concienciada de la magnitud en la que la prematuridad e FGR, tema principal de trabajo, afecta a una gran cantidad de personas, las cuales ven su vida completamente cambiada. Dado que muchos de los neonatos que nacen prematuros presentarán secuelas durante su vida como problemas de salud, aprendizaje, desarrollo, etc., las cuales se extienden a los padres y demás familiares, y teniendo en cuenta que 1 de cada 13 nacimientos en España es prematuro [2], alrededor de 28.000 anuales [3], es muy importante estudiar a fondo este tema con el fin de encontrar una explicación e intentar mejorar la situación.</p> <p>Se utilizará la metodología CRISP-DM [4], ya que es comúnmente utilizada en el desarrollo de este tipo de proyectos y que divide el proceso de Data Mining en 6 fases principales: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación e Implantación.</p> <p>Como metodología de gestión del tiempo se utilizará Agile, se ha construido un Kanban en el que se hará seguimiento semanal.</p> <p>Los resultados y conclusiones del proyecto se presentarán en la memoria final.</p>	
Abstract (in English, 250 words or less):	
<p>The objective of this project is to detect patterns in the data provided by BCNatal Fetal Medicine Research Center to discover if any the hidden relationships between Intra-Uterine Growth Restriction (FGR) [1], and prematurity -the two most significant causes of neonatal morbidity-, and how these relationships can help early detection.</p> <p>The motivation to develop the project is personal, since I think that society is not sufficiently aware of the magnitude in both prematurity and FGR, the main topic of work, affects a large number of people, who see their lives completely changed. Since many of the babies born prematurely will have sequelae during their life such as health problems, learning, development,</p>	

etc., which extend to parents and people closed to them, and taking into account that 1 in 13 births in Spain is Premature [2], around 28,000 annually [3], it is very important to study this topic thoroughly in order to find an explanation and try to improve the situation.

The CRISP-DM methodology [4] will be used, since it is commonly used in the development of this kind of projects and that divides the Data Mining process into 6 main phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Implementation.

An Agile methodology will be used to manage tasks and timing (Kanban).

The results and conclusions of the project will be presented in the final report.

Índice

1.	Introducción.....	9
1.1.	Contexto y justificación del trabajo	9
1.2.	Objetivos secundarios del trabajo	9
1.3.	Enfoque y método seguido.....	10
1.4.	Planificación del trabajo.....	10
1.4.1.	Descripción Recursos necesarios para el desarrollo del trabajo.....	10
1.4.2.	Planificación temporal de cada tarea.....	10
1.5.	Alcance del proyecto	11
1.6.	Breve resumen de los productos obtenidos	11
2.	Estado del arte o análisis de mercado	12
2.1.	Antecedentes.....	12
2.2.	Resolución del problema en la actualidad	13
2.3.	Objetivo	13
2.4.	Metodología	14
2.5.	Factores	14
2.6.	Resultados	15
2.7.	Conclusiones.....	16
2.8.	Justificación ámbito Técnico	16
3.	Diseño e implementación del trabajo	17
3.1.	Análisis y limpieza de datos	17
3.1.1.	Identificación de variables objetivo	17
3.1.2.	Exploración y limpieza de datos.....	17
3.1.2.1.	Comprobación de veracidad de los datos.....	17
3.1.2.2.	Limpieza de variables.....	18
3.1.2.2.1.	Limpieza de variables categóricas.....	18
3.1.2.2.2.	Limpieza de variables numéricas	18
3.1.2.3.	Visualización de variables	18
3.1.2.3.1.	Visualización de variables categóricas	18
3.1.2.3.2.	Visualización de variables numéricas.....	18
3.1.3.	Estudio de los valores nulos.....	19
3.2.	Definición de estudios a realizar	19
3.2.1.	Introducción.....	19
3.3.	Búsqueda de variables significativas: Búsqueda de fenotipos identificando variables con importancia, predictores perfectos y variables no relevantes, para la predicción temprana de problemas en el feto.....	21
3.3.1.	Técnica ANOVA	21
3.3.2.	Agrupación de variables por significado	22
3.3.3.	Fusión de grupos de variables para el estudio.....	23
3.4.	Modelado	23
3.4.1.	Introducción.....	23
3.4.2.	Modelado de estudios a realizar.....	25
3.4.2.1.	Predicción temprana de problemas en el feto o nacimiento de neonato pequeño: Mediante el uso de variables que se puedan obtener dentro de los 2 primeros trimestres de embarazo.....	25

3.4.2.1.1.	Estudio 1: Predicción del percentil de crecimiento en el nacimiento (Predicción de la variable BW_Percentile).....	25
3.4.2.1.1.1.	Técnica XGBoost	25
3.4.2.1.2.	Estudio 2: Predicción de feto con problemas (nacimiento de neonato pequeño) (Predicción de variable 'BW_Percentile_inferior_10)	26
3.4.2.1.2.1.	Técnica XGBoost	26
3.4.2.1.2.2.	Técnica Regresión Logística	27
3.4.2.1.2.3.	Técnica Red neuronal simple	27
3.4.2.1.3.	Estudio 3: Predicción de feto con problemas y grado de FGR (Predicción variables BW_Percentile_inferior_10 y FGR_birth_nivel)	28
3.4.2.1.3.1.	Técnica XGBoost	28
3.4.2.1.3.2.	Técnica Regresión Logística	29
3.4.2.1.3.3.	Técnica Red neuronal simple	29
3.4.2.1.4.	Estudio 4: Predicción de grado de FGR (Predicción variables FGR_birth_nivel)	30
3.4.2.1.4.1.	Técnica XGBoost	30
3.4.2.1.4.2.	Técnica Regresión Logística	30
3.4.2.1.4.3.	Técnica Red neuronal simple	30
3.4.2.2.	Predicción no temprana de problemas en el feto o nacimiento de neonato pequeño: Incluyendo también en el estudio las variables obtenidas en el tercer trimestre de gestación. ...	30
3.4.2.2.1.	Estudio 1: Agregar al modelo las variables de la ecografía Doppler	32
3.4.2.2.2.	Estudio 2: Agregar al modelo las variables perfiladas del estudio 1 y las variables de la ecografía fetal del tercer trimestre	32
4.	Experimentos, validación y resultados	32
4.1.	Introducción.....	32
4.2.	Predicción temprana de problemas en el feto o nacimiento de neonato pequeño: Mediante el uso de variables que se puedan obtener dentro de los 2 primeros trimestres de embarazo.	33
4.2.1.	Estudio 1: Predicción del percentil de crecimiento en el nacimiento (Predicción de la variable BW_Percentile).....	33
4.2.1.1.	Técnica XGBoost	33
4.2.2.	Estudio 2: Predicción de feto con problemas (nacimiento de neonato de variable 'BW_Percentile_inferior_10).....	34
4.2.2.1.	Técnica XGBoost	34
4.2.2.1.1.	Búsqueda mejores hiperparámetros GridSearCV() y modelo con observaciones con nulos corregidas.....	34
4.2.2.1.2.	Eliminar observaciones con nulos.....	35
4.2.2.2.	Técnica Regresión logística	36
4.2.2.2.1.	Eliminar observaciones con nulos.....	36
4.2.2.2.2.	Corregir observaciones con nulos.....	36
4.2.2.3.	Técnica Red neuronal simple	36
4.2.2.3.1.	Búsqueda mejores hiperparámetros GridSearCV() y modelo con observaciones con nulos corregidas.....	36
4.2.2.3.2.	Eliminar observaciones con nulos.....	36

4.2.3.	Estudio 3: Predicción de feto con problemas y grado de FGR (Predicción variables BW_Percentile_inferior_10 y FGR_birth_nivel)	37
4.2.3.1.	Técnica XGBoost	37
4.2.3.1.1.	Eliminar observaciones con nulos.....	37
4.2.3.1.2.	Corregir observaciones con nulos.....	37
4.2.3.2.	Técnica Regresión logística	38
4.2.3.2.1.	Eliminar observaciones con nulos.....	38
4.2.3.2.2.	Corregir observaciones con nulos.....	38
4.2.3.3.	Técnica Red neuronal simple	38
4.2.3.3.1.	Eliminar observaciones con nulos.....	38
4.2.3.3.2.	Corregir observaciones con nulos.....	38
4.2.4.	Estudio 4: Predicción de grado de FGR (Predicción variables FGR_birth_nivel)	38
4.2.4.1.	Técnica XGBoost	39
4.2.4.1.1.	Eliminar observaciones con nulos.....	39
4.2.4.1.2.	Corregir observaciones con nulos.....	39
4.2.4.2.	Técnica Regresión logística	39
4.2.4.2.1.	Eliminar observaciones con nulos.....	39
4.2.4.2.2.	Corregir observaciones con nulos.....	39
4.2.4.3.	Técnica Red neuronal simple	39
4.2.4.3.1.	Eliminar observaciones con nulos.....	39
4.2.4.3.2.	Corregir observaciones con nulos.....	39
4.3.	Predicción no temprana de problemas en el feto o nacimiento de neonato pequeño: Incluyendo también en el estudio las variables obtenidas en el tercer trimestre de gestación.....	40
4.3.1.	Estudio 1: Agregar al modelo las variables de la ecografía Doppler	40
4.3.1.1.	Agregar variables al modelo	40
4.3.1.2.	Perfilado de las variables del Estudio 1.....	41
4.3.2.	Estudio 2: Agregar al modelo las variables perfiladas del estudio 1 y las variables de la ecografía fetal del tercer trimestre	42
4.3.2.1.	Agregar variables al modelo	42
4.3.2.2.	Perfilado de las variables del Estudio 2.....	43
4.4.	Procedimiento para unir 3 modelos en uno único (output)	43
4.5.	Procedimiento para pasar nuevos datos al modelo (output)	44
4.6.	Resumen de Fenotipos y modelos seleccionados.....	45
4.6.1.	Modelo 1: Predicción temprana de feto con problemas y grado de FGR (Unión de 2 modelos)	45
4.6.1.1.	Selección del modelo.....	45
4.6.1.2.	Fenotipos del modelo	46
4.6.1.3.	Significado de las variables seleccionadas	46
4.6.2.	Modelo 2: Predicción no temprana de feto con problemas y grado de FGR con variables de la ecografía Doppler (Unión de 2 modelos).....	47
4.6.2.1.	Selección del modelo.....	47
4.6.2.2.	Fenotipos del modelo	47

4.6.2.3.	Significado de las variables seleccionadas	48
4.6.3.	Modelo 2: Predicción no temprana de feto con problemas y grado de FGR con variables de la ecografía Doppler y ecografía fetal del tercer trimestre (Unión de 2 modelos).....	48
4.6.3.1.	Selección del modelo.....	48
4.6.3.2.	Fenotipos del modelo	49
4.6.3.3.	Significado de las variables seleccionadas	49
5.	Estudio adicional: Predicción del diagnóstico de los médicos sobre estimación de feto con problemas y grado de gravedad.....	50
5.1.	Predicción temprana de juicio de los médicos del nivel de gravedad del feto (variable Fetal_problem_1)	50
5.2.	Predicción temprana de juicio de los médicos de feto con problemas (variable Fetal_problem_before_delivery).....	51
5.3.	Predicción del grado de gravedad (predicción variable Fetal_problem_1)	52
5.3.1.	Agregar variables perfiladas al modelo (Estudio 1).....	52
5.3.2.	Agregar variables perfiladas al modelo (Estudio 2).....	52
5.4.	Predicción de juicio de los médicos de neonato nacido con problemas (variable Fetal_problem_before_delivery).....	53
5.4.1.	Agregar variables perfiladas al modelo (Estudio 1).....	53
5.4.2.	Agregar variables perfiladas al modelo (Estudio 2).....	53
5.5.	Estudio comparativo de aciertos entre juicio de los médicos y el modelo construido en este trabajo	53
5.5.1.	Modelo construido en este estudio (Apartado 4.6.3).....	53
5.5.2.	Diagnóstico de los médicos (Variable Fetal_problem_before_delivery).....	54
6.	Líneas de trabajo futuras.....	54
7.	Conclusiones.....	55
8.	Glosario	56
9.	Bibliografía	58

1. Introducción

1.1. Contexto y justificación del trabajo

Punto de partida del trabajo

El objetivo principal de este proyecto es detectar patrones en los datos ofrecidos por BCNatal Fetal Medicine Research Center con el fin de averiguar si la relación entre los ellos es determinante o explicativa de la Restricción de Crecimiento Intra-Uterino (FGR) [1], una de las causas más significativas de morbilidad neonatal, para tratar de predecir de manera temprana esta restricción.

¿Por qué es un tema relevante?

Dado que muchos de los neonatos que nacen prematuros presentarán secuelas durante su vida como problemas de salud, aprendizaje, desarrollo, etc., las cuales se extienden a los padres y demás familiares, y teniendo en cuenta que 1 de cada 13 nacimientos en España es prematuro, alrededor de 28.000 anuales [2], es muy importante estudiar a fondo este tema con el fin de encontrar una explicación e intentar mejorar la situación.

¿Cómo se resuelve el problema de momento?

Actualmente, para tratar de detectar el FGR en el feto, los médicos llevan a cabo alguna/s de las siguientes técnicas:

- Realización de diferentes tipos de ecografías. [5]
- Monitorización fetal para evaluar la frecuencia cardíaca y los movimientos del feto. [5]
- Pruebas de cribado para detectar posibles infecciones. [5]
- Amniocentesis para descartar causas genéticas de la restricción del crecimiento intrauterino. [5]

Cuando el FGR [1] ya se ha diagnosticado, los médicos recomiendan que la madre siga un control más exhaustivo acudiendo a revisiones más frecuentes durante el resto de su embarazo. [5]

Aportación realizada o resultado deseado

El resultado que se desea obtener en este trabajo es predecir con un cierto grado de confianza si existe relación significativa entre la información de los fenotipos de los neonatos diagnosticados con FGR, con el fin de intentar pronosticar la posible ocurrencia futura o de detectarlos precozmente.

1.2. Objetivos secundarios del trabajo

- Identificar qué información de los fenotipos del set de datos explican el nacimiento de un neonato pequeño y/o con FGR.
- Utilizar esta información para construir un modelo que ayude, en la medida de lo posible, a la predicción temprana de estas condiciones.
- Utilizar esta información para construir otro modelo que ayude, en la medida de lo posible, a la predicción de estas condiciones, dentro de un período de tiempo más próximo al parto, un período aproximado de 1 a 2 meses.
- Detectar distintos tipos de FGR para mejorar el diagnóstico y pronóstico de los neonatos.

1.3. Enfoque y método seguido

Para desarrollar este proyecto, al inicio del mismo se previó utilizar la metodología CRISP-DM [4], ya que es comúnmente utilizada en el desarrollo de este tipo de estudios y que divide el proceso de Data Mining en 6 fases principales: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación e Implantación. Sin embargo, finalmente no se ha aplicado ninguna metodología estándar, el proceso seguido ha sido un ciclo de análisis y exploración de los datos, limpieza, modelado, evaluación e inicio de un nuevo ciclo.

Como metodología de gestión del tiempo se utilizará una metodología Agile, para lo cual se ha construido un Kanban en el que se hará seguimiento semanal durante el transcurso del proyecto, sin embargo, dada la situación de Covid-19 presente durante el desarrollo del proyecto, el kanban inicial ha sido modificado. Esta información se muestra en el siguiente apartado 1.4. *Planificación del trabajo*.

Se parte de una base de datos en bruto, la cual no ha sido tratada previamente.

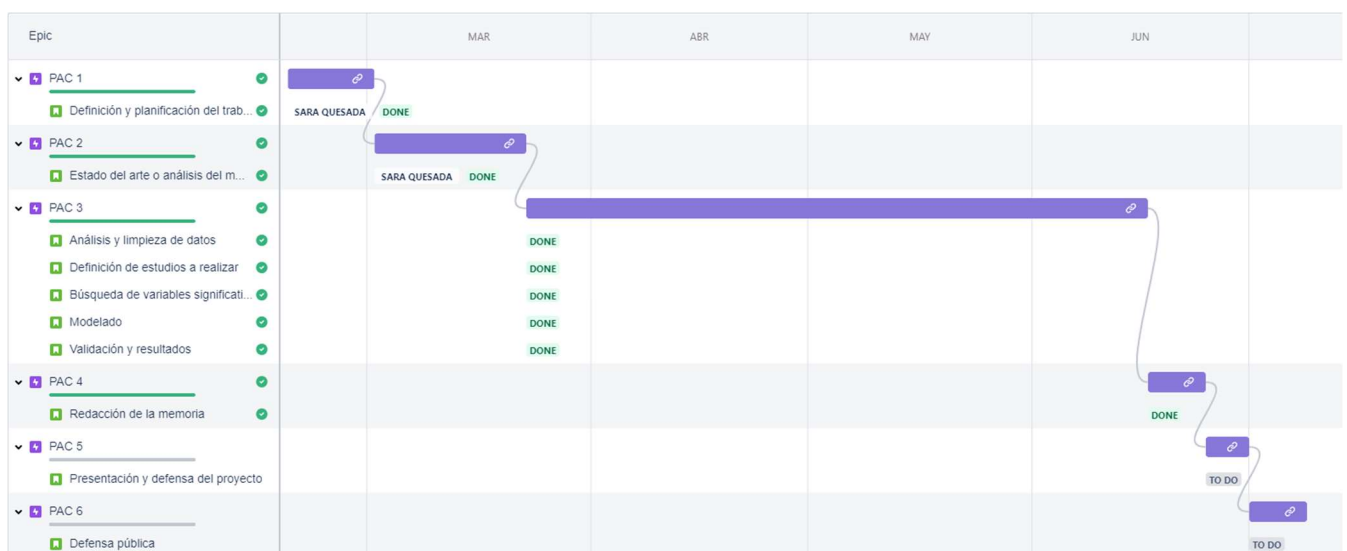
1.4. Planificación del trabajo

1.4.1. Descripción Recursos necesarios para el desarrollo del trabajo

- Set de datos proporcionado por BCNatal Fetal Medicine Research Center.
- Para llevar a cabo el desarrollo, se utilizará un ordenador con procesador AMD Ryzen 5 2600 Six-Core 3.40 GHz con 16Gb de RAM y con la versión 3.7.1 de Python instalada.
- Para la gestión de tareas y soporte documental se utilizará el paquete Atlassian, concretamente los módulos Jira y Confluence, respectivamente.
- Número de horas de dedicación semanal: 50 horas.

1.4.2. Planificación temporal de cada tarea

Se muestra la planificación de cada tarea en un Kanban desarrollado en Jira:



Como se puede comprobar, al seguir una metodología Agile, no se planificó un tiempo para las tareas de la PAC 3. Una nueva tarea se inicia al finalizar la anterior.

1.5. Alcance del proyecto

El alcance del proyecto es encontrar los fenotipos que predigan de manera prematura si un feto nacerá pequeño y/o presentará FGR, mediante el set de datos de entrada que contiene datos pre y post natales, y construir un modelo para aplicarlo a nuevos datos futuros. No se prevé utilizar los datos post natales ya que el objetivo es la predicción temprana, no ver la evolución posterior del neonato.

Asimismo, se persigue el mismo objetivo, pero añadiendo al estudio datos no tan precoces, sino más próximos a la fecha de parto, dentro de un período de entre 30 y 60 días de antelación al nacimiento, para intentar construir un modelo con predicción no tan prematura, pero posiblemente más precisa.

1.6. Breve resumen de los productos obtenidos

Esta información también se puede encontrar en el siguiente enlace de Github: https://github.com/squesagi/FGR_fenotipos.git

Los productos que se obtendrán al final del desarrollo del proyecto son los siguientes:

- FGR_fenotipos.ipynb: Fichero con el código del desarrollo del proyecto en lenguaje Python.
- Phenomapping_original.xlsx: Juego de datos de entrada (Solicitar a Elisenda Bonet ebonetca@uoc.edu)
- Memoria.docx: Documento de extensión .docx que describe el desarrollo del proyecto.
- Anexo.docx: Documento de extensión .docx que complementa el apartado 2 “Estado del arte o análisis de mercado” de la memoria.
- Memoria.pdf: Documento de extensión .pdf que describe el desarrollo del proyecto.
- Anexo.pdf: Documento de extensión .pdf que complementa el apartado 2 “Estado del arte o análisis de mercado” de la memoria.
- Imagenes: Carpeta que contiene los diagramas de los árboles XGBoost que, debido a su gran tamaño, se han agregado como enlace después de cada ejecución de técnica XGBoost para que se pueda hacer un zoom y ver al completo su contenido.
- Modelos: Carpeta que contiene los modelos construidos a lo largo del proyecto. Concretamente se construye los siguientes 3 modelos:
 - Modelo 1: Predicción prematura de nacimiento de neonato con o sin FGR y grado de FGR: Unión de los modelos **xgboost_model1** y **xgboost_model2**.
 - Modelo 2: Predicción no prematura de nacimiento de neonato con o sin FGR y grado de FGR (Prueba Ecografía Doppler): Unión de los modelos **xgboost_model_vars_periodico1** y **xgboost_model_vars_periodico2**.
 - Modelo 3: Predicción prematura de nacimiento de neonato con o sin FGR y grado de FGR (Prueba ecografía fetal tercer trimestre): Unión de los modelos **xgboost_model_vars_periodico_3trim2_1** y **xgboost_model_vars_periodico_3trim2_2**.

En el apartado 3 se explicará el porqué de esta unión de 2 modelos y en el apartado 4 se expondrá detalladamente cómo se realiza la carga de los modelos.

2. Estado del arte o análisis de mercado

Como se indicó en el apartado anterior, el objetivo de este proyecto es averiguar si la relación entre los datos de estudio es determinante o explicativa de la restricción de Crecimiento Intrauterino y la prematuridad, tratando de construir un modelo que ayude a la detección temprana de estas dos patologías que desmoronan por completo la vida de millones de familias.

Para construir el estado del arte, el cual representa la revisión bibliográfica que existe sobre el tema de trabajo motivo de este proyecto, se ha realizado una selección de las investigaciones que se han elaborado sobre el tema de trabajo, y se tratará de extraer conclusiones y encontrar relaciones entre las evidencias descubiertas por los autores en sus estudios.

Se ha decidido estructurar el estado del arte en los siguientes apartados, los cuales se describirán de manera global ofreciendo una idea genérica del conjunto de todas las fuentes, además de citar cada una de ellas cuando se mencione alguno de sus puntos de estudio, para poder detectar rápidamente en qué temas coincidieron los autores de los artículos y no hacer más denso este apartado:

- Antecedentes: Este apartado se utilizará para entrar en contexto y explicar resumidamente qué se sabe o qué conocimiento hay del tema.
- Objetivo: Se resumirá el objetivo principal de los autores en la realización de su estudio.
- Metodología: Se describirá el procedimiento que los autores han llevado a cabo para realizar su estudio.
- Factores: Se indicarán las causas que hacen que se dé el objeto tema de estudio. Por ejemplo, ¿qué factores hacen que se dé el FGR?
- Resultados: Este apartado contendrá las evidencias a las que han logrado llegar los autores de los artículos a través de sus estudios.
- Conclusiones: En esta sección se añadirán las conclusiones de los autores después de realizar sus estudios.
- Justificación ámbito técnico: Se razonará el motivo de selección de las metodologías y herramientas que se utilizarán en el proyecto.

2.1. Antecedentes




La restricción de crecimiento intrauterino, es una de las principales causas de morbilidad y mortalidad perinatal [11] que afecta a entre el 5 y 10% de todos los neonatos [11,17]. Asimismo, entre un 7 y un 10% de los nacimientos son prematuros [17,18,19], siendo alrededor de un 75% los ingresos en Unidades de Cuidados Intensivos de neonatos por prematuridad [18].

Hoy en día, todavía por parte de los médicos se intenta concienciar a las mujeres embarazadas, o que desean embarazarse, de lo importante que es cuidarse y mantener una vida sana para evitar los posibles y probables problemas futuros que tendrá su neonato, los cuales en muchos de los casos son irreversibles y afectarán durante su edad adulta, conduciéndoles incluso a la muerte [8,9,11,22]. Asimismo, además del coste emocional que afrontarán las familias, el coste económico que conlleva el cuidado de estos neonatos es elevado, ya que no sólo requerirán estar en la Unidad de Cuidados Intensivos de Neonatos en incubadoras, sino que constantemente se les harán todo tipo de pruebas para monitorizar su evolución [8].

Se cuenta con la premisa de que la mayoría de los factores que conducen a enfermedades cardiovasculares crónicas ya están presentes en la infancia, pero todavía se debe indagar en este campo. [23]

2.2. Resolución del problema en la actualidad

Actualmente, muchos de los autores explican en sus estudios que el proceso que siguen para tratar el problema es el siguiente [8,9,10,11,12,13,14,15,16,17,20,23]:

- Control prenatal con exploración física.
- Ecografías focalizadas en la cabeza, abdomen y fémur del feto, con lo que se calculará un peso fetal estimado, el cual se comparará con las estadísticas en base al número de semanas de gestación para estimar el percentil de crecimiento.
- Pruebas de cribado del primer trimestre: Se hacen para comprobar si el feto tiene riesgos de sufrir una anomalía cromosómica como síndrome de Down o una malformación congénita. Esta prueba se realizará en la semana 12 de embarazo, máximo en la 14, ya que si se supera esta fecha la prueba ya no tendrá ningún valor. En esta prueba también se analizarán marcadores sanguíneos como las hormonas Gonadotropina coriónica y la Proteína plasmática unida al embarazo, y se combinarán con la edad de la madre. Con todo esto se calculará el riesgo que tiene el feto con un intervalo confianza de entre el 80 y 90%.
- Ecografía Doppler fetal para medir los flujos en los vasos sanguíneos del feto para obtener diferentes evidencias. Por ejemplo, a través de la medición de los flujos de la arteria umbilical, se puede determinar si la placenta está funcionando correctamente. También, a través de la medición de los vasos sanguíneos del cerebro del feto para ver si este está creciendo adecuadamente y/o identificar en qué etapa de FGR se encuentra.
- Analizar la etapa de FGR en la que se encuentra el feto (4 etapas). Una vez clasificado el feto como feto de riesgo por su bajo crecimiento, existen dos maneras diferentes de proceder en este caso, según el riesgo del feto:
 - Si en cada control de crecimiento los vasos fetales se van alterando más en relación a su flujo  Feto con alto riesgo de contraer enfermedades, o incluso de muerte dentro del útero  Se inducirá el parto a partir de una determinada semana de gestación, aunque implique un parto prematuro.
 - Si el riesgo no es alto  Se realizarán controles de crecimiento del feto.

2.3. Objetivo

Entre los objetivos de los estudios de los autores de la bibliografía seleccionada destacan los siguientes:

- Concienciar e informar a las madres o futuras madres de la importancia y repercusión del FGR y resolver la posible falta de información [8,9,11,22].
- Intentar prevenir el FGR, o detectarlas lo antes posible, estableciendo relaciones entre las variables que ayuden a explicar por qué el feto no crece de la manera adecuada [8,9,10,11,12,13,16,21,23].

- Intentar prevenir los partos prematuros tratando de detectar a tiempo los posibles factores que los generan [14,15,18,19,21].
- Identificar diferentes fenotipos de FGR [24].
- Estudiar otras enfermedades que tienen en común con el FGR la dificultad en detectar de manera temprana la enfermedad, como por ejemplo la enfermedad de Alzheimer, lo cual es vital para tratar de paliar sus efectos. [27,28,29].

2.4. Metodología

Se realizan diferentes estudios para tratar de lograr los objetivos, en los cuales se comprueba que muchos de los autores coinciden:

- Se evalúan los efectos de los hábitos maternos en el feto [8,9,10,11,12,13,14,15,16,17,20,23]. De manera concreta, se hace un estudio sobre la posible efectividad de la actividad física materna durante el embarazo en el FGR [20].
- Se estudia el impacto del grupo de edad de la madre en la prematuridad [19].
- Se investiga la asociación entre la morbilidad neonatal prematura, el desarrollo infantil y el desarrollo escolar [21].
- Se aplican los modelos basados en eventos para tratar de evaluar la progresión de la enfermedad [27,28,29,30].
- Se aplican otros métodos de aprendizaje supervisado y no supervisado para tratar de diferenciar diferentes tipos de fenotipos de FGR [24,25,26].

2.5. Factores

Según los autores, los factores o causas que hacen que se dé el FGR se dividen en 3 grandes grupos:

a) Factores maternos:

- Problemas de nutrición [8,9,10,13,20].
- Preeclampsia (causa más común que supone el 10% de los FGR) [8,9,10,12,13].
- Hábitos no saludables (consumo de tabaco, alcohol, ciertos fármacos, teratógenos o estupefacientes) [9,10,13,16,20,21].
- Infecciones [9,10,12].
- Constitución pequeña [10].
- Privación social, factor económico [10,20,21].
- Genética [11].
- Enfermedades de tipo autoinmune, especialmente, el Síndrome Antifosfolípido y trombofilias, lupus eritematoso sistémico, o problemas renales [12,13].
- Trastornos depresivos [13].
- Estado civil [20,21].
- Diabetes [20].
- Atención prenatal [20,21].
- Edad [21].

b) Factores fetales:

- Malformaciones fetales [10,13,16].
- Infecciones [10,13,16].
- Prematuridad [11,16].
- Cromosomopatías [13,16].

- Genopatías [13,16].
- Embarazos múltiples [13,16].
- Anemia [13,16].

c) Factores placentarios:

- Problemas en la placenta [8,9,10,11,16].
- Placenta circunvalada [13].
- Infartos placentarios [13].
- Vasculitis [13].
- Arteria umbilical única [13].
- Inserción velamentosa del cordón [13].
- Tumores placentarios [13].
- Angiogénesis aberrante [13].

Los factores estudiados más comúnmente y que se relacionan con un riesgo incrementado de sufrir un parto prematuro son:

- FGR.
- Infecciones urinarias [14].
- Alteraciones en el desarrollo de la gestación [14].
- Situación o situaciones anormales en el desarrollo del cuello uterino (Causa principal) [14].
- Edad de la madre [18].
- Embarazos múltiples [18].
- Hábitos no saludables (consumo de tabaco, alcohol, ciertos fármacos, teratógenos o estupefacientes) [18].
- Problemas de salud de la madre [18].

2.6. Resultados

Mediante el desarrollo de sus estudios, los autores obtuvieron las siguientes evidencias en las cuales se puede comprobar que hay autores que coinciden en las mismas o que llegaron al mismo resultado:

- FGR puede afectar a 1 o 2 de cada 10 embarazos [8,17,18].
- Una de las principales causas de FGR son problemas en la placenta [8,10,12].
- Las alteraciones en la curva de crecimiento durante las primeras etapas de formación del feto, conllevan un alto riesgo de desarrollo de enfermedades crónicas a lo largo de la vida del neonato [11,13,15,16,21,23].
- Los neonatos nacidos con bajo peso, durante su vida adulta, tienen tendencia a desarrollar obesidad, hipertensión y diabetes, entre otras enfermedades [11,13,23].
- Las mujeres demasiado delgadas tienen 4 veces más posibilidades de tener un neonato de bajo peso, que una madre de peso normal [11,16].
- Las mujeres embarazadas que realizaron ejercicio aeróbico regularmente durante su embarazo mostraron una menor incidencia de macrosomía fetal (peso superior a 4000 gramos al nacer) y diabetes gestacional que aquellas que no lo hicieron [20].
- Se obtiene una evidencia de relación entre FGR, la remodelación cardíaca y la disfunción longitudinal en la infancia, que aumentan linealmente con el nivel de gravedad de FGR [23].
- Se aplica el método SNF en dos estudios diferentes. En uno se consigue diferenciar correctamente 3 fenotipos que se asemejan a FGR temprano, tardío y SGA tardío (pequeño para la edad gestacional), y en el otro se logra distinguir a 3 grupos de pacientes [24,25].

2.7. Conclusiones

Los autores llegaron a las siguientes conclusiones a través de sus estudios:

Se concluye que lo más importante es detectar el FGR lo antes posible, ya que mediante la aplicación de ciertos tratamientos se puede, por ejemplo, intentar madurar los órganos para que se desarrollen lo máximo posible, lo que probablemente mejoraría el estado del feto, hecho muy importante teniendo en cuenta que una proporción significativa de los prematuros probablemente tengan enfermedades crónicas a lo largo de su vida [11,12].

El efecto de la actividad física resulta ser beneficioso para el crecimiento fetal, y por lo que el impacto del ejercicio aeróbico puede resultar influyente de manera positiva en la tolerancia a la glucosa [20].

FGR no figura entre las condiciones que se presume que aumentan el riesgo cardiovascular según las variables de estudio, aunque se logra proporcionar evidencia de una asociación entre el FGR y la remodelación cardíaca y la disfunción longitudinal en la infancia que muestra un aumento lineal con la gravedad de la restricción del crecimiento y es independiente de la edad gestacional en el parto, perfil lipídico, o índice de masa corporal [23].

La programación cardíaca primaria podría ser una de las causas del aumento de la mortalidad cardiovascular en adultos nacidos con FGR, y esto puede abrir nuevas oportunidades de monitoreo e intervención en recién nacidos y niños afectados con esta afección [23].

Los modelos de aprendizaje automático no supervisados (el método SNF) generaron 3 fenotipos que se asemejan a FGR temprano, tardío y SGA tardío. Los resultados proporcionan nuevos conocimientos sobre la base molecular y la heterogeneidad clínica de los FGR. Respaldan el uso de modelos de aprendizaje automático para avanzar en la comprensión de la naturaleza multifenotípica (múltiples fenotipos) de los FGR [24].

El método SNF funciona correctamente en la búsqueda de fenotipos de FGR y SGA y en la detección de tumores en pacientes, cuando en su aplicación se combinan datos genéticos, micro ARNs y análisis de variación del número de copias [24,25].

2.8. Justificación ámbito Técnico

Se ha estudiado la aplicación del modelo basado en eventos para la progresión de la enfermedad que se puede usar para estimar el orden temporal de los cambios neuropatológicos a partir de datos transversales [26,27,28,29,30]. Aunque el campo en el que se ha visto el estudio no es FGR, el hecho de estudiar otras enfermedades que tienen en común con el FGR la dificultad en detectar de manera temprana la enfermedad, por ejemplo, de Alzheimer, hace que pueda resultar muy interesante aplicar métodos para descubrir patrones en el tema principal del proyecto, que han funcionado en una enfermedad que es difícil de detectar.

También se ha estudiado la aplicación del método no supervisado SNF, mediante el cual los diferentes autores han obtenido resultados óptimos de agrupación, especialmente se considera interesante que se hayan obtenido muy buenos resultados de diferenciación de fenotipos de FGR y SGA, pequeño para la edad gestacional, razón suficiente para aplicarlo a este proyecto [24,25].

La metodología que se ha seleccionado para organizar el desarrollo del proyecto ha sido CRISP-DM, ya que se adapta perfectamente a un proyecto de minería de datos al dividirse en 6 fases de comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implantación, muy próximo al proceso natural de Data Mining [4,31].

Se ha seleccionado el paquete Atlassian para la gestión de tareas y soporte documental, ya que sigue una metodología agile la cual permitirá que se completen satisfactoriamente todas las fases del proyecto, además de facilitar el despliegue y disponer de soporte documental para realizar seguimiento. Mediante el artículo, se demuestra que este paquete se adapta perfectamente a las necesidades [34].

Asimismo, además de probar a aplicar en los datos de estudio el método descubierto SNF y los modelos basados en eventos, también entre los métodos a probar estarán el método de regresión lineal múltiple, ya que el objetivo es encontrar relaciones entre los datos que expliquen el FGR y la prematuridad, además del Análisis de Componentes Principales (PCA), ya que se espera recibir una gran cantidad de variables de estudio [32,33].

3. Diseño e implementación del trabajo

El desarrollo de este apartado se encuentra en el fichero *"FGR_fenotipos.ipynb"*, adjunto a la documentación de entrega, donde el nivel de detalle es mayor que en la memoria, ya que esta está limitada a 60 páginas.

3.1. Análisis y limpieza de datos

Como se irá viendo durante el desarrollo de este proyecto, uno de los retos del set de datos de entrada es el elevado número de valores nulos que presenta este. Esto obligará a descartar muchas variables que podrían haber sido muy descriptivas del problema que lamentablemente no se podrán utilizar, ya que, por el gran volumen de nulos, la variable no aportará valor.

Asimismo, también se comprobará que hay variables que contienen valores nulos que realmente no lo serán, simplemente toman valor 0 ya que están relacionadas con el valor de otra variable. Todos estos ejemplos se irán analizando y corrigiendo a lo largo del proyecto.

A continuación, se muestra una captura con información de los datos al inicio del estudio:

```
El set de datos tiene 878 variables y 1245 observaciones.  
Nº de valores nulos en la variable "BW_Percentile": 59.  
El set de datos tiene 878 variables y 1186 observaciones.  
El dataset contiene 553 observaciones de madres de las cuales los médicos previeron que presentarían problemas fetales antes de  
1 parto, y 685 que se previó que no presentarían problemas.  
Nº de variables con valores nulos: 863  
Nº de variables sin valores nulos: 15
```

Se identifica la variable 'BW_Percentile' como variable objetivo, ya que indica el percentil con el que nació el neonato.

3.1.1. Identificación de variables objetivo

Se identifica la variable 'BW_Percentile' como variable objetivo, ya que contiene el percentil de nacimiento del neonato, con valores del 0 al 100. Para facilitar la predicción, se crea la variable 'BW_Percentile_inferior_10' a partir de la variable 'BW_Percentile' que indicará si el neonato nació con un percentil inferior al 10 o no (posibilidad de FGR y/o SGA).

También se crea la variable 'FGR_birth_nivel' para indicar si el neonato nació pequeño y su grado de severidad, siendo 0 para los valores de la variable 'BW_Percentile' superiores a 10, 'Leve' para los valores entre 10 y 5, 'Moderado' para los valores entre 5 y 2, y 'Severo' para los valores inferiores a 2, detectados durante la fase de 'Estudio del arte' e incluidos en la bibliografía [13]. Así, se realizará el estudio en base a la predicción de estas 3 variables, siendo la variable 'BW_Percentile_inferior_10' el nivel mínimo de zoom y 'BW_Percentile' el máximo. Sin embargo, se ha creado la variable 'FGR_birth_nivel' que identifica no sólo si un neonato nacerá pequeño, sino que también determina el grado de severidad en este caso.

3.1.2. Exploración y limpieza de datos

3.1.2.1. Comprobación de veracidad de los datos

De todas las variables de tipo DateTime, se comprobarán los valores mínimo y máximo de esa variable, para verificar que la fecha sea correcta, por ejemplo, que no indique una fecha de realización de pruebas muy antigua o superior a la fecha actual.

3.1.2.2. Limpieza de variables

3.1.2.2.1. Limpieza de variables categóricas

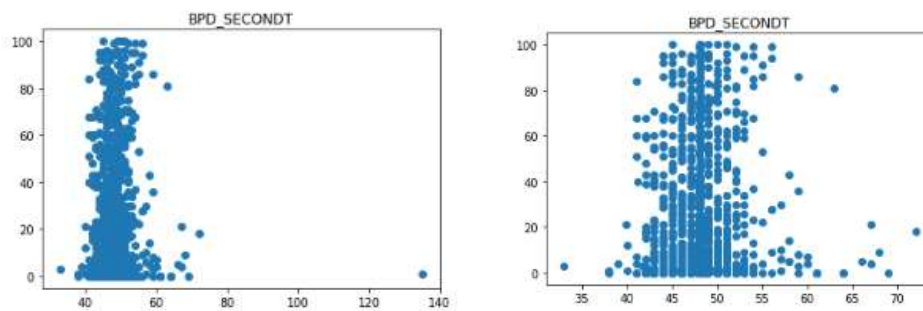
Se crearán una serie de métodos que se reutilizarán con el fin de limpiar las variables, ya que en el juego de datos existen muchos valores que corresponden a la misma categoría, pero que se han registrado con errores o en otro idioma. También se generalizan algunas de las categorías de las variables que se prevé utilizar (más adelante se realizará un estudio de los valores nulos). Se creará un nuevo dataframe según el tipo de datos de las variables y se detectarán las variables categóricas disfrazadas de enteros o floats, y a la inversa. Asimismo, se identificará qué variables representan ID's. Estas variables no aportarán valor al estudio, por lo que se excluirán del estudio si cumplen que sólo contienen valores únicos.

3.1.2.2.2. Limpieza de variables numéricas

Para construir el dataframe de variables cuantitativas, se identificarán para descartar las variables categóricas, las de tipo Datetime y las de tipo ID. Para tratar de identificar los outliers, se establecerá el siguiente criterio y después de obtener el resultado, se analizará una a una cada variable, mediante el método describe(), para comprobar si realmente se tratan de outliers. El criterio establecido es el siguiente:

- Outliers < $Q1 - 1.5 \cdot IQR$
- Outliers > $Q3 + 1.5 \cdot IQR$

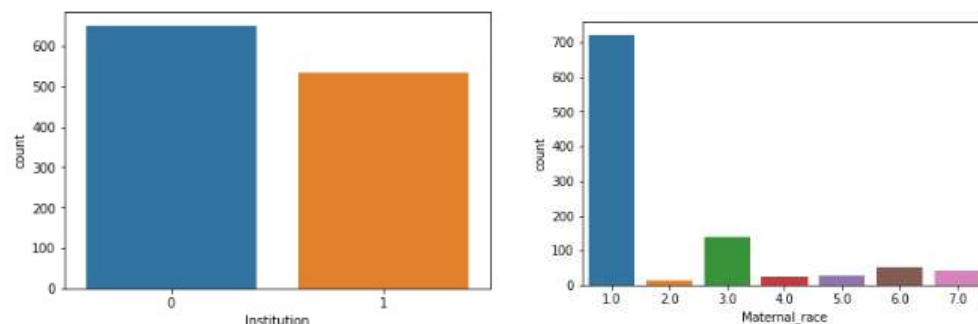
A continuación, se muestra un ejemplo de corrección de outlier, en un diagrama de dispersión, antes y después de corregirlo:



3.1.2.3. Visualización de variables

3.1.2.3.1. Visualización de variables categóricas

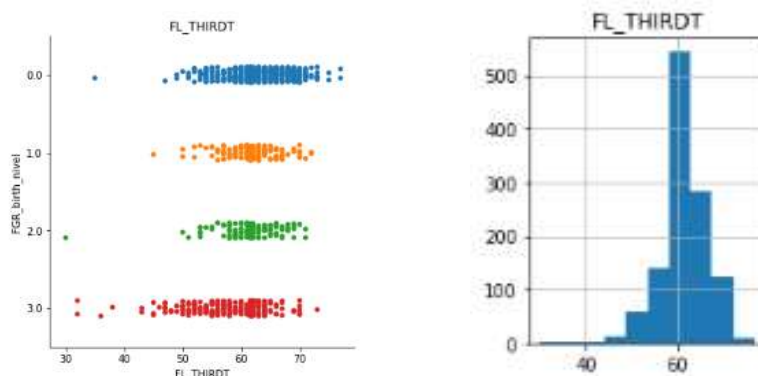
Se estudia la distribución de las variables categóricas con un diagrama de barras, donde se verá que hay variables que están equilibradas en cuanto a número de observaciones por categoría, y muchas otras en las cuales la distribución es muy diferente. Se muestran un par de ejemplos:



3.1.2.3.2. Visualización de variables numéricas

Para estudiar la distribución de las variables numéricas, se realizará mediante diagramas de dispersión individuales, de los cuales se ha mostrado un ejemplo en el apartado 3.1.2.2.2.,

también mediante diagramas de dispersión en función de una de las variables objetivo 'FGR_birth_nivel' y mediante histogramas. A continuación, se muestra un ejemplo de una variable donde existían valores que habían sido detectados como posibles outliers, los cuales se dispersan del resto de valores, pero los cuales finalmente se ha comprobado que eran valores correctos debido a que eran observaciones de neonatos que nacieron con restricción de crecimiento:



3.1.3. Estudio de los valores nulos

Como se ha comentado anteriormente, una de las limitaciones del proyecto es que el juego de datos de entrada contiene una gran cantidad de nulos, de los cuales se han corregido varios. Sin embargo, como se verá en adelante, muchas variables interesantes y que seguramente aportarían mucho valor al estudio deberán ser descartadas del mismo, ya que es insostenible trabajar con ellas debido a la gran cantidad de valores nulos. Para realizar un análisis exhaustivo, se crea el dataframe 'df_NaN' el cual contendrá el nombre de cada variable, el número total de nulos, el porcentaje de nulos respecto al total de observaciones del juego de datos y el tipo de datos de la variable:

```

Nº de variables con valores nulos: 798
Nº de variables sin valores nulos: 85

```

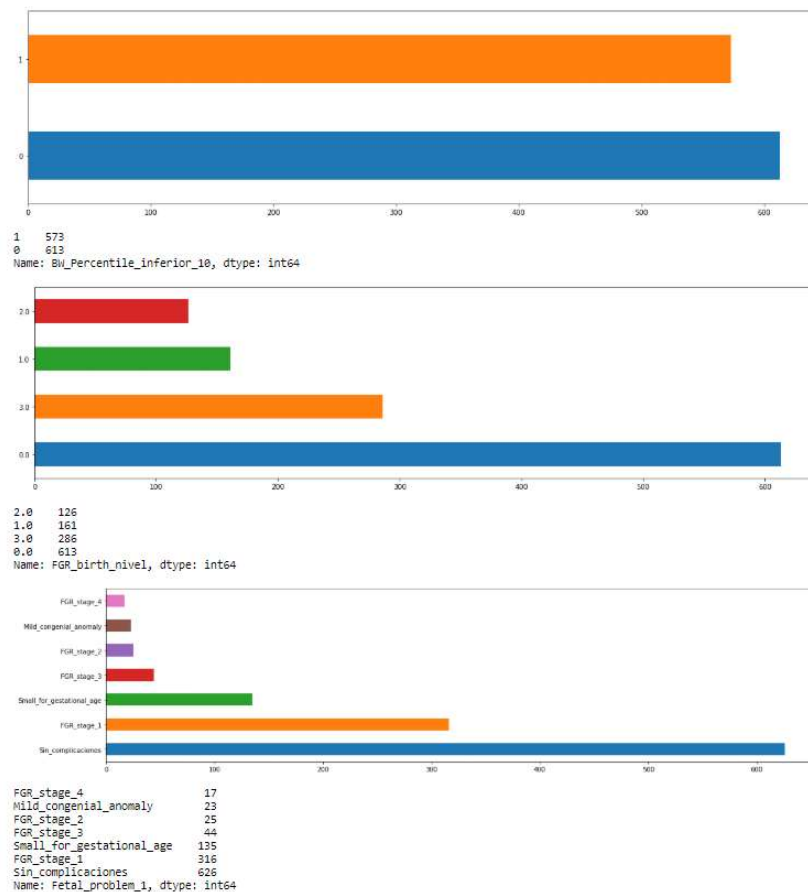
	Variables	Nulls	%_Nulls	Tipo_datos
0	Date_DX_GH	1176	99.16	datetime64[ns]
0	Name_medication	1174	98.99	category
0	Fetal_problem_2	1171	98.74	category
0	Hair_PTpgmg	1168	98.48	float64
0	Calcarine	1165	98.23	float64

Si se compara el número de variables con valores nulos al inicio de este apartado, se podrá comprobar que se ha pasado de tener 15 variables sin ningún nulo a 85.

3.2. Definición de estudios a realizar

3.2.1. Introducción

Para tratar de alcanzar los objetivos definidos en el apartado 1, se comenzará mostrando la distribución de las variables objetivo. La variable 'BW_Percentile' se analizará a través de las variables 'BW_Percentile_inferior_10' y 'FGR_birth_nivel', las cuales se crearon a partir de ella. Asimismo, se agrega al estudio la variable 'Fetal_problem_1', pero desde otra perspectiva de investigación, ya que esta variable describe la predicción que realizaron los médicos y el estudio únicamente se realizará como un agregado al proyecto.



Analizando el primer diagrama de barras, se puede ver que las observaciones de cada categoría están bastante equilibradas, lo que significa que la muestra comprende aproximadamente el mismo número de observaciones de fetos pequeños de tamaño o con restricción, que sanos.

Al analizar el segundo diagrama, se ve que existe un problema de distribución en los datos que, si no se trata, puede conducir fácilmente a una situación de overfitting. Por ejemplo, si se analiza la clase '0', se puede ver que el número de observaciones es superior a la suma de las observaciones de las demás categorías (consecuencia del reparto equilibrado de la muestra entre fetos con problemas y sin problemas). Esto significa que los valores 0 están sobreajustando el modelo, de manera que de cada 1 vez que se entrena el modelo para los valores '2', el modelo se entrena 4,86 veces para los valores '0' (613/126). Esto sucede también en el tercer diagrama. Para tratar esto, es necesario equilibrar los casos. Para ello se realiza el siguiente planteamiento el cual afectará a los estudios que se realizarán con las variables objetivo 'FGR_birth_nivel' y 'Fetal_problem_1' (predicciones de los médicos), pero no a los estudios de la variable 'BW_Percentile_inferior_10', ya que se ha comprobado que la muestra está compensada:

- a) **OPCIÓN 1:** El output será la conjunción de los siguientes 2 modelos:
- **Modelo 1:** En este modelo únicamente se intentará predecir los valores 0 y 1, es decir, si el neonato nace pequeño o no.
 - **Modelo 2:** Se descartan del juego de datos los valores 0, ya que están sobreajustando el modelo y en el modelo 1 ya se habrán predicho, y además únicamente se utilizará el mismo número de observaciones de la clase con menos observaciones de las categorías (en el caso de la variable 'FGR_birth_nivel', la categoría '2' es la que menos clasificaciones tiene, en total 126), seleccionando para ello ese mismo número de observaciones (126) de entre toda la muestra para cada categoría. Es decir, si se dispone de 126 observaciones en la categoría con menos observaciones, de entre todas las observaciones de las otras clases, únicamente se seleccionarán 126 de cada una de las categorías para entrenar el modelo, pero se testeará con el conjunto total de test, evitando así el posible overfitting.

- b) **OPCIÓN 2:** Se realizará el mismo desarrollo que en el modelo 2 de la OPCIÓN 1, pero en un único modelo sin separar los valores '0'.

De este modo, estratificando la muestra, se reducirá el riesgo de overfitting de los modelos que se construyan. A continuación, se definirán los estudios a realizar en los próximos apartados:

- **ESTUDIO 1:** Predicción del percentil de crecimiento en el nacimiento (Predicción de la variable BW_Percentile)
- **ESTUDIO 2:** Predicción de feto con problemas (nacimiento de neonato pequeño) (Predicción de variable 'BW_Percentile_inferior_10')
- **ESTUDIO 3:** Predicción de feto con problemas y grado de FGR (Unión de 2 modelos: Predicción de variable 'BW_Percentile_inferior_10' y 'FGR_birth_nivel')
- **ESTUDIO 4:** Predicción de grado de FGR (Predicción de variable 'FGR_birth_nivel' en un único modelo)

Con la información obtenida de estos estudios se creará un modelo para la predicción temprana, y otro para la predicción no temprana, se enfocarán desde 2 perspectivas diferentes:

- a) **Predicción temprana de problemas en el feto o nacimiento de neonato pequeño:** Se construirá un modelo mediante el uso de variables que se puedan obtener dentro de los 2 primeros trimestres de embarazo.
- b) **Predicción de problemas en el feto o nacimiento de neonato pequeño:** Se construirá un modelo a partir del modelo anterior, Incluyendo también en el estudio las variables obtenidas en el tercer trimestre de gestación. En el apartado 3.4. se comprobará qué grupos de variables se utilizarán.

Adicionalmente, se aplicará el mismo estudio a la variable 'Fetal_problem_1' para tratar identificar los fenotipos que indujeron a los médicos a realizar su diagnóstico previo al parto, y finalmente se comparará si acierta más el juicio de los médicos o el modelo creado en este proyecto.

3.3. Búsqueda de variables significativas: Búsqueda de fenotipos identificando variables con importancia, predictores perfectos y variables no relevantes, para la predicción temprana de problemas en el feto

3.3.1. Técnica ANOVA

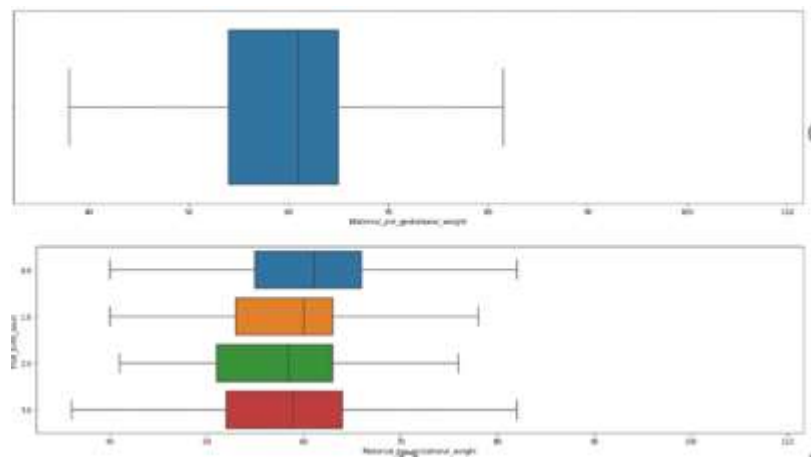
Para dar respuesta al objetivo del proyecto, se intentará detectar qué variables pueden predecir, con un cierto grado de confianza, si un feto sufrirá problemas o por el contrario el neonato nacerá dentro de un percentil igual o superior a 10.

Para comenzar, se aplicará la técnica Anova para identificar qué variables explican más varianza del conjunto de datos, con un grado de confianza del 95%. Debido a que no hay muchas observaciones en el conjunto de datos, se puede aplicar un ANOVA teniendo en cuenta todas las variables numéricas del juego de datos, contra la variable 'FGR_birth_nivel'. Asimismo, se crea una función que a la vez que se vaya aplicando el Anova, seleccione las variables en las que exista una diferencia estadísticamente significativa en las medias, es decir, $p\text{-value} < 0.05$, según la condición fijada del 95% de confianza. Los nombres de estas variables se guardarán en la lista "numéricas_explicativas" y a continuación se les hará un estudio de los nulos. A continuación, se muestra un ejemplo de aplicación de Anova a una variable:

```
variable: Maternal_pre_gestational_weight
df      sum_sq  mean_sq      F    PR(>F)
Y      3.0    1122.199796  274.000599  3.200093  0.022617
Residual 1182.0  138128.494613  116.899925      NaN      NaN

Se rechaza la hipótesis nula (p-value: 0.0226 < 0.05) y se asume que las medias de las distintas poblaciones son diferentes. Existe una diferencia estadísticamente significativa entre las medias.

Maternal_pre_gestational_weight
FGR_birth_nivel
0.0      62.621207
1.0     59.930825
2.0     58.704206
3.0     60.394196
```



A continuación, de estas variables explicativas se seleccionarán las que tengan más del 70% de los datos, y se creará el dataframe “df_data_numericas_70”. Estas variables son:

```
df_data_numericas_70: Index(['EFW_THIRDT', 'PERCENTIL_EFW_THIRDT', 'Second_trim_Hto', 'AC_DX',
'BPD_DX', 'FL_DX', 'PERCENTIL_EFW_DX', 'EFW_DX',
'GA_US_Diagnosis_Fetal_Suboptimal_Growth',
'Maternal_age_at_the_time_of_recruitment', 'Garecruitmentcalculated',
'Maternal_pre_gestational_weight', 'SBP_at_recruitment',
'DBP_at_recruitment', 'HC_DX', 'IP_MAut_DX', 'Second_trim_Hb',
'IP_ACM_DX', 'GA_US_Third_trim', 'BPD_THIRDT', 'HC_THIRDT', 'AC_THIRDT',
'FL_THIRDT', 'IP_AU_DX', 'AC_SECOND'],
dtype='object')
```

En relación a las variables categóricas, se seleccionan las que tengan más del 70% de los datos, y se crea el dataframe “df_data_categoric_70”.

```
df_data_categoric_70: Index(['Institutional_or_extra_institutional_US', 'Gestational_diabetes',
'Previous_preg_complications', 'Drugs', 'Previous_term_gestations',
'Live_births', 'Abortions', 'Previous_preterm_gestations', 'Alcohol',
'Nulliparity', 'Smoke', 'GDM', 'Magnesium_Sulfate_neuroprotection',
'Gestational_hypertension', 'PE', 'Steroids_during_gestation',
'Other_preg_Problems', 'Route_of_delivery', 'Education_mother',
'Maternal_race', 'Fetal_echoardiography', 'Maternal_work_status',
'Paternal_race', 'Education_father', 'Paternal_work_status',
'Previous_medical_history', 'Pregnancy_concibed_through_IVF',
'Specific_Maternal_Disease_Bool', 'p3a', 'p2a1b', 'p2a2a', 'p2a2b',
'p2b1a', 'p2b1b', 'p2b1c', 'p2b1d', 'p2b2', 'p3b', 'p1b3c', 'p3c',
'p3d', 'Mat_angiogenic', 'Continent_of_birth_mother', 'p2a1a', 'p1b3b',
'Gender', 'p1a3b', 'Continent_of_birth_father', 'Placenta_pathology',
'p1a1a', 'p1a1b', 'p1a2a1', 'p1a2a2', 'p1a2b', 'p1a3a', 'p1a3c',
'p1b3a', 'p1b1a', 'p1b1b', 'p1b1c', 'p1b2a1', 'p1b2a2', 'p1b2a3',
'p1b2b1', 'p1b2b2', 'Institution'],
dtype='object')
```

3.3.2. Agrupación de variables por significado

Paralelamente, se decide realizar un estudio para buscar las variables que, según su significado, puedan ser importantes para el estudio. Esta investigación se ha incorporado como anexo en el propio fichero “FGR_fenotipos.ipynb”, sección “Anexo 1: Estudio de importancia de variables por agrupación” para no hacer más denso el código principal. Para realizar el estudio, se crearán dataframes para analizar por grupos de variables, que han sido agrupadas según su significado:

```
df_sociodemografico_padres1 = data.iloc[:,6:19]
df_sociodemografico_padres2 = data.iloc[:, [880,881]]
df_sociodemografico_padres = pd.concat([df_sociodemografico_padres1,df_sociodemografico_padres2],axis=1)
df_habitos_maternos = data.iloc[:, [26,28,29]]
#41 es la variable objetivo de 'df_datos_embarazos_anteriores':
df_datos_embarazos_anteriores = data.iloc[:, [30,31,32,33,34,41,45,47,48]]
df_datos_embarazo_actual1 = data.iloc[:, [50,51,53,54,56,57,58,59,60,70,71,73,74,153,155,157,158,161,162,336,340,341]]
df_datos_embarazo_actual2 = data.iloc[:,369:403]
df_datos_embarazo_actual = pd.concat([df_datos_embarazo_actual1,df_datos_embarazo_actual2],axis=1)
df_ultrasonido_doppler_1 = data.iloc[:,91:96]
df_ultrasonido_doppler_2 = data.iloc[:,145:149]
df_ultrasonido_fetal_2_trim = data.iloc[:, [62,64,65,66,67,68,69]]
df_ultrasonido_fetal_3_trim = data.iloc[:,76:83]
df_ultrasonido_fetal_entre_2_y_3_trim = data.iloc[:, [84,85,86,87,88,89,90,97]]
df_biomarcadores_parto = data.iloc[:, [164,167,168,172,177,178,185,187]]
df_biomarcadores_nutricionales = data.iloc[:,188:220]
df_calidad_somnolencia1 = data.iloc[:,222:233]
df_calidad_somnolencia2 = data.iloc[:,234:244]
df_calidad_somnolencia = pd.concat([df_calidad_somnolencia1,df_calidad_somnolencia2],axis=1)
df_estres_ansiedad = data.iloc[:,244:284]
df_estres_percibido = data.iloc[:,284:298]
df_patologia_placenta1 = data.iloc[:,298:336]
df_patologia_placenta2 = data.iloc[:,337:340]
df_patologia_placenta = pd.concat([df_patologia_placenta1,df_patologia_placenta2],axis=1)
df_biomarcadores_sangre_materna1 = data.iloc[:,343:369]
df_biomarcadores_sangre_materna2 = data.iloc[:,403:451]
df_biomarcadores_sangre_materna = pd.concat([df_biomarcadores_sangre_materna1,df_biomarcadores_sangre_materna2],axis=1)
#Observaciones relacionadas en 'df_metabolomica_sangre_materna', 'df_metabolomica_cordon_umbilical' y 'df_metabolomica_fetal':
df_metabolomica_sangre_materna = data.iloc[:,474:477]
df_metabolomica_cordon_umbilical = data.iloc[:,523:549]
df_metabolomica_fetal = data.iloc[:,588:636]
```


Si se consulta el estudio en “Anexo 1: Estudio de importancia de variables por agrupación”, se podrá comprobar que las variables con más importancia del estudio son las siguientes, las cuales se almacenan en el dataframe “df_representativos”:

```
df_representativos: Index(['Maternal_age_at_the_time_of_recruitment', 'Paternal_age', 'Drugs',
'Smoke', 'SBP_at_recruitment', 'DBP_at_recruitment',
'OSullivan_test_result', 'Second_trim_Hb', 'First_trim_PAPP_A_MOM',
'DBP_First_trim', 'SBP_First_trim', 'First_trim_Hto', 'CRL',
'Down_Syndrome_Risk', 'IPm_UTA_First_trim', 'PE', 'Metab_2018',
'IP_AU_DX', 'IP_ACM_DX', 'IP_mAut_DX', 'Aortic_Isthmus_DX',
'IP_AU_LastDoppler', 'IP_ACM_LastDoppler', 'IP_DV_LastDoppler',
'GA_US_Second_trim', 'PERCENTIL_EFW_SECOND', 'EFW_SECOND',
'AC_SECOND', 'PERCENTIL_EFW_THIRD', 'EFW_THIRD', 'PERCENTIL_EFW_DX',
'GG', 'Q2_Minutes_get_sleep', 'PSS7', 'PSS10', 'Weight',
'ratiofetoplacenta', 'Mat_LPhenylalanine_1', 'Maternal_VLDL_TG',
'Maternal_PLGF', 'Fetal_Small_HDL_P_(umol/L)', 'Fet_Aceticacid_1'],
dtype='object')
```

3.3.3. Fusión de grupos de variables para el estudio

En este momento, se dispone de los siguientes grupos de datos:

- df_data_numericas_70
- df_data_categoric_70
- df_representativos

Además, se crea la lista “variables_descartadas_estudio”, con las variables identificadas como predictores perfectos, variables objetivo, explicativas de datos posteriores al parto, o variables que únicamente indican si se realizó o no una prueba:

```
variables_descartadas_estudio: ['Survey_12_months', 'ASQ_One_Year', 'Emergency_C_section', 'inducccion', 'Stillbirth', 'Place_o
f_birth_mother', 'Occupation_mother', 'Place_of_birth_father', 'Occupation_father', 'Fetal_problem_before_delivery', 'Specific_Fe
tal_Problem', 'Fetal_problem_1', 'BW_Percentile_inferior_10', 'FGR_birth_nivel', 'GA_at_birth_days', 'Specific_Maternal_Diseas
e', 'Study_ID', 'Hospital_ID', 'BW_Percentile', 'Gadeliverycalculated', 'Birthweight', 'GA_at_birth-weeks', 'GA_at_birth-days',
'Metab_PILOT', 'Metab_2018', 'Emergency_C_section', 'inducccion', 'Apgar_test_1', 'Apgar_test_5', 'Stillbirth', 'Route_of_delive
ry', 'BW_Percentile_inferior_10', 'FGR_birth_nivel', 'DateofDelivery', 'Fetal_problem_before_delivery', 'Recruitment_date', 'Sp
ecific_Maternal_Disease', 'GA', 'Route_of_delivery', 'GA_at_birth_weeks']
```

A continuación, se realiza un cruce de información y se obtienen los 2 siguientes grupos de datos con los que se podrá modelar:

- **variables_destacadas_utilizar:** Contiene la unión de las variables contenidas en 'df_data_numericas_70', 'df_data_categoric_70' y 'df_representativos', descartando las variables contenidas en 'variables_descartadas_estudio'.
- **variables_representativas_utilizar:** Contiene las variables contenidas en 'df_representativos', descartando las variables contenidas en 'variables_descartadas_estudio'.

Se ha realizado esta distinción de grupos, para analizar la incidencia de modelar teniendo en cuenta únicamente las variables del estudio del apartado 3.3.2. o teniendo en cuenta el resto de variables seleccionadas.

3.4. Modelado

3.4.1. Introducción

Al tratar un problema de clasificación, se decide seleccionar las siguientes 3 técnicas estadísticas para modelar:

- **TÉCNICA XGBOOST:** Se selecciona este árbol ya que se trata de una técnica de clasificación muy robusta.
- **TÉCNICA REGRESIÓN LOGÍSTICA:** Se selecciona la regresión logística ya que las diferentes variables objetivo son categóricas.
- **TÉCNICA RED NEURONAL SIMPLE:** Se selecciona la red neuronal para probar evaluar si esta técnica ofrece buenos resultados en los estudios.

Para facilitar la selección de los datos y hacer el proceso más eficiente, se construyen una serie de funciones las cuales se irán reutilizando durante el desarrollo del proyecto.

Para este, se utilizarán los dos conjuntos de datos `'df_variables_destacadas_utilizar'` y `'df_variables_representativas_utilizar'`, pero los cuales se han perfilado mediante su aplicación a las técnicas de XGBoost, regresión logística y red neuronal simple, probando a predecir las variables `'BW_Percentile_inferior_10'` y `'FGR_birth_nivel'` y comparando la importancia de cada variable en los modelos, comparando la incidencia de añadir o descartar cada variable en el mismo, comparando precisión, etc. y siempre dando prioridad al objetivo del proyecto de predecir de manera temprana si el feto sufrirá problemas, es decir, si el neonato nacerá pequeño para la edad gestacional o con FGR, sacrificando así la precisión en la predicción a favor de obtener una predicción temprana del problema.

Este ciclo no se ha incorporado a este código debido a que se haría extremadamente denso y se basa en, como se ha mencionado, la aplicación de las variables de entrada contenidas en `'df_variables_destacadas_utilizar'` y `'df_variables_representativas_utilizar'` a las técnicas del apartado 3.4.2., básicamente se ha realizado un perfilado de las variables.

Como se ha comentado, ese ha sido el desarrollo del perfilado de las variables hasta que finalmente se han obtenido las siguientes variables que, como se verá posteriormente, explican y ayudan a identificar si el neonato nace pequeño y los diferentes tipos de FGR con cierta precisión, logrando así el objetivo principal de este proyecto. Las variables encontradas en los dos grupos de datos, se han agrupado en un único grupo:

```
variables_seleccionadas_estudio_FGR: Index(['SBP_at_recruitment', 'Second_trim_Hto', 'Second_trim_Hb',
      'DBP_at_recruitment', 'Maternal_pre_gestational_weight',
      'GA_US_Second_trim', 'Placenta_pathology', 'EFW_SECOND',
      'Maternal_age_at_the_time_of_recruitment', 'First_trim_Hto',
      'AC_SECOND', 'PE', 'Smoke', 'Steroids_during_gestation',
      'OSullivan_test_result', 'DBP_First_trim', 'SBP_First_trim'],
      dtype='object')
```

De este modo, los fenotipos que ayudarán a lograr el objetivo se encuentran en el dataframe `"variables_seleccionadas_estudio_FGR"` reduciendo a únicamente 17 variables, y con obtención dentro de los dos primeros trimestres, de las casi 900 de entrada, facilitando así la obtención de los datos para la predicción de futuros casos.

A continuación, se analizan los nulos de las variables del dataframe `"variables_seleccionadas_estudio_FGR"`:

```
Variables y nº de nulos:
SBP_First_trim          400
DBP_First_trim          400
OSullivan_test_result   284
Second_trim_Hto         243
First_trim_Hto          241
Smoke                   166
Maternal_age_at_the_time_of_recruitment  122
Steroids_during_gestation  28
dtype: int64
```

De las variables categóricas, únicamente contienen nulos `'Smoke'` y `'Steroids_during_gestation'`. Se muestra la distribución en las categorías para analizar la manera en que se tratarán sus nulos durante la fase de modelado:

```
0.0    805
1.0     215
Name: Smoke, dtype: int64

0.0    1021
1.0     137
Name: Steroids_during_gestation, dtype: int64
```

Analizando los valores nulos y la distribución de las categorías con nulos se plantea lo siguiente:

a) Eliminar las observaciones con nulos.

b) Corregir los valores nulos:

- Variables numéricas: Para limpiar los valores numéricos, estos se sustituirán por la mediana que es resistente a outliers.
- Variables categóricas: Se asignará una categoría aleatoria de entre las categorías de la variable: De este modo se conseguirá equilibrar el reparto y no dar más peso a una categoría que a otra.

Se crea el método '**asignar_mediana_y_o_clase_aleatoria(dataset,y_column)**' para tratar los nulos. Se verá el resultado de aplicar estos dos planteamientos al modelizar, donde se analizará la diferencia de precisión entre aplicar una alternativa u otra, y si la diferencia no es exageradamente a favor de eliminar los nulos, siempre se priorizará la sustitución de los nulos por la mediana y categoría aleatoria de la variable, que la eliminación.

Asimismo, durante el perfilado de datos, se identifica el siguiente grupo de variables muy explicativas de entre todos los datos, las cuales, como se ha mencionado en el punto anterior, no se tendrán en cuenta para este primer estudio, donde se pretende predecir a la mayor brevedad la posibilidad de restricción en el feto, pero sí para el estudio del apartado 3.4.2.2. donde se añaden al modelo las variables que se han detectado con mayor varianza, pero las cuales se ha decidido no aplicar en este estudio debido a que son variables que se obtienen de pruebas realizadas en el tercer trimestre de embarazo (con una media de 40 días de antelación al parto), tiempo de oro que permitiría aplicarle los tratamientos oportunos al feto para que paliase o solucionase sus posibles problemas, sin embargo, como se ha comentado, el objetivo principal es la predicción temprana del problema. Estas variables son:

- **Grupo de variables 'altamente explicativas'**: ['PERCENTIL_EFW_DX', 'IP_mAut_DX', 'IP_ACM_DX', 'BPD_THIRDT', 'HC_THIRDT', 'AC_THIRDT', 'FL_THIRDT', 'EFW_THIRDT', 'PERCENTIL_EFW_THIRDT', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX', 'HC_DX', 'AC_DX', 'FL_DX', 'EFW_DX', 'PERCENTIL_EFW_DX', 'IP_AU_DX', 'Aortic_Isthmus_DX', 'IP_DV_DX', 'AFI_DX', 'MVP_mm_DX', 'Mat_angiogenic', 'GA_Areacruitmentcalculated', 'IP_mAut_DX', 'GA_US_Third_trim']

3.4.2. Modelado de estudios a realizar

Como se comentó en el apartado anterior, para tratar los nulos, cada estudio se realizará desde 2 perspectivas diferentes:

- a) Eliminar las observaciones con nulos.
- b) Corregir los valores nulos con la mediana para las variables numéricas, o con un valor aleatorio entre las categorías de la variable, para las variables categóricas.

En este apartado de modelado, se mostrará únicamente la arquitectura utilizada en la sección 4, pero no los resultados, ya que estos se mostrarán en el apartado 4.

3.4.2.1. Predicción temprana de problemas en el feto o nacimiento de neonato pequeño: Mediante el uso de variables que se puedan obtener dentro de los 2 primeros trimestres de embarazo

3.4.2.1.1. Estudio 1: Predicción del percentil de crecimiento en el nacimiento (Predicción de la variable BW_Percentile)

En este estudio se intentará predecir precozmente el percentil de crecimiento al nacer, es decir, con datos obtenidos dentro de los 2 primeros trimestres de embarazo.

3.4.2.1.1.1. Técnica XGBoost

Se aplicará la técnica XGBoost, con la opción de eliminar observaciones con nulos:

```

# 'BW_Percentile'
# Modelo 1: Predecir variable 'BW_Percentile':
print('*****Modelo 1*****')

# Parámetros:
y_column = 'BW_Percentile'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR, y], axis=1)
y = dataset[y_column]
muestra_estratificada = False
model_parameters = {
    'max_depth': random.randint(2, 15),
    'n_estimators': random.choice([20, 30, 40]),
    'learning_rate': random.choice([0.01, 0.05, 0.1])
}
grid_parameters = {'n_jobs': 10, 'cv': 10}

# Eliminar las observaciones con nulos:
features, y = eliminar_observaciones_nulos(dataset, y_column)

# Se quieren utilizar las categorías, por esto se convierten a dummies:
features = convertir_a_dummies(features)

# Para obtener el mejor modelo según los parámetros pasados y además graficar sus resultados, se llama a la función
xgboost_model = get_xgboost_model_individual(features, y, y_column, model_parameters)

```

Se aplicará la técnica XGBoost, con la opción de corregir observaciones con nulos:

```

# 'BW_Percentile'
# Modelo 2: Predecir variable 'BW_Percentile':
print('*****Modelo 2*****')

# Parámetros:
y_column = 'BW_Percentile'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR, y], axis=1)
y = dataset[y_column]
muestra_estratificada = False
model_parameters = {
    'max_depth': random.randint(2, 15),
    'n_estimators': random.choice([20, 30, 40]),
    'learning_rate': random.choice([0.01, 0.05, 0.1])
}
grid_parameters = {'n_jobs': 10, 'cv': 10}

# Asignar la mediana de la variable a las variables numéricas y una categoría aleatoria de entre las categorías
features, y = asignar_mediana_y_o_clase_aleatoria(dataset, y_column)

# Se quieren utilizar las categorías, por esto se convierten a dummies:
features = convertir_a_dummies(features)

# Para obtener el mejor modelo según los parámetros pasados y además graficar sus resultados, se llama a la función
xgboost_model = get_xgboost_model_individual(features, y, y_column, model_parameters)

```

Como se puede comprobar, únicamente cambia una línea de código entre un modelo y otro, la línea de llamar al método “eliminar_observaciones_nulos()” o la de llamar al método “asignar_mediana_y_o_clase_aleatoria()”. En este apartado 3.4.2. únicamente se mostrará la arquitectura del modelo llamando a la función “asignar_mediana_y_o_clase_aleatoria()” para no alargar innecesariamente este documento, y ya que únicamente cambia esa línea de código mencionada.

3.4.2.1.2. Estudio 2: Predicción de feto con problemas (nacimiento de neonato pequeño) (Predicción de variable 'BW_Percentile_inferior_10')

En este estudio se intentará predecir precozmente si el neonato nació pequeño (inferior al percentil 10) o no, es decir, con datos obtenidos dentro de los 2 primeros trimestres de embarazo.

3.4.2.1.2.1. Técnica XGBoost

Para buscar los mejores parámetros del modelo, se aplicará la técnica GridSearchCV():

```

# OBTENER MEJORES PARÁMETROS:
# 'BW_Percentile_inferior_10'
# Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR, y], axis=1)
muestra_estratificada = False
model_parameters = {
    'max_depth': [2, 4, 8, 10],
    'n_estimators': [10, 20, 30, 40],
    'learning_rate': [0.001, 0.01, 0.05, 0.1]
}

grid_parameters = {'n_jobs': 10, 'cv': 10}

features, y = asignar_mediana_y_o_clase_aleatoria(dataset, y_column)
features = convertir_a_dummies(features)

# Para obtener el mejor modelo según los parámetros pasados y además graficar sus resultados, se llamará a la función
best_parameters_XGBoost_model1, best_model, X_train, X_test, y_train, y_test = xgboost_model_grid(features, y,
model_parameters, grid_parameters, muestra_estratificada, y_column)

# A continuación, con los parámetros obtenidos de llamar al método xgboost_model_grid(), se llamará a la función
xgboost_best_model

# Para graficar resultados y calcular algunos coeficientes del mejor modelo obtenido en el grid:
xgboost_best_model = best_xgboost_model_individual(best_model, X_train, X_test, y_train, y_test)

```

A partir de este punto, para aplicar la técnica XGBoost, se partirá de los mejores parámetros que se hayan encontrado en la sección 3.4.2.1.2.1., los cuales se mostrarán en el apartado 4. A continuación, se aplicará la técnica XGBoost, con la opción de corregir observaciones con nulos:

```
#'BW_Percentile_inferior_10'
#Modelo 3: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('*****Modelo 3*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
xgboost_model = get_xgboost_model_individual(features, y, y_column, model_parameters)
```

3.4.2.1.2.2. Técnica Regresión Logística

Se aplicará la técnica Regresión logística, con la opción de corregir observaciones:

```
#REGRESIÓN LOGÍSTICA:
#Modelo 2: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
reg_logistica_model_2 = get_regresion_logistica_model_individual_multiclass(features, y, y_column, muestra_estratificada)
```

3.4.2.1.2.3. Técnica Red neuronal simple

Para buscar los mejores parámetros del modelo, se aplicará la técnica GridSearchCV(). Después de hacer numerosas pruebas con diferentes parámetros, se llega a la conclusión de que la red no aprende y aunque se le han pasado al grid diferentes parámetros los resultados son los mismos, aun cambiando la arquitectura de la red (número de capas densas, dropout, etc.) por lo que se decide únicamente reducir a una o dos capas de 256 y 128 neuronas respectivamente, 10,30 y 50 épocas, 0.1, 0.01 y 0.001 valor para learning rate. También se probó con agregar al grid hasta 6 capas con diferentes neuronas, lo que consumía mucha memoria y paraba la ejecución del notebook por falta de memoria, por lo que fue necesario ir probando las combinaciones de parámetros de pocas en pocas para evitar la falta de memoria, pero en cualquier situación la precisión seguía siendo muy baja.

```
#RED NEURONAL SIMPLE - GRID
#Búsqueda de mejores parámetros con GridSearchCV:

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False
binaria = True
#se descartan más épocas, más capas y más neuronas porque la red no aprende
layers = [[256],[128,128]]
epochs=[10,30,50]
lr = [0.1,0.01,0.001]
batch_size = [64]
param_grid = dict(layers=layers, epochs=epochs, lr=lr, batch_size=batch_size)
grid_parameters = {'n_jobs': 1, 'cv': 10}

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)

#Llamar a la función para construir el modelo:
df_grid_models_rnn, best_model_rnn, best_parameters_rnn = rnn_model_grid(features, y, param_grid, grid_parameters, muestra_estratificada, y_column)

#Llamar al mejor modelo:
rnn_model_1 = get_rnn_model_individual(features, y, y_column, best_parameters_rnn['layers'], best_parameters_rnn['lr'], best_parameters_rnn['epochs'], best_parameters_rnn['batch_size'], binaria)

df_grid_models_rnn.head()
```

Se aplicará la técnica de Red neuronal simple, con la opción de corregir observaciones con nulos:

```
#Construcción modelo de red neuronal:
#Modelo 2: Predecir variable 'BW_Percentile_inferior_10'.
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False
layers = best_parameters_rnn['layers']
lr = best_parameters_rnn['lr']
epochs = best_parameters_rnn['epochs']
batch_size = best_parameters_rnn['batch_size']
binaria = True

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
rnn_model_3 = get_rnn_model_individual(features, y, y_column, layers, lr, epochs, batch_size, binaria)
```

3.4.2.1.3. Estudio 3: Predicción de feto con problemas y grado de FGR (Predicción variables BW_Percentile_inferior_10 y FGR_birth_nivel)

En este estudio se intentará predecir precozmente si el neonato nació pequeño y su grado de FGR. Para ello se creará un modelo para la primera opción y otro para la segunda, y el output será la unión de los dos modelos, tal y como se mencionó en el apartado 3.2. Los datos a utilizar serán los obtenidos dentro de los 2 primeros trimestres de embarazo.

3.4.2.1.3.1. Técnica XGBoost

```
#1. 'BW_Percentile_inferior_10'
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
xgboost_model1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'FGR_birth_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
#esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
#La clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
dataset2 = dataset2[dataset2['FGR_birth_nivel']!=0]
dataset2['FGR_birth_nivel'] = dataset2['FGR_birth_nivel'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
xgboost_model2 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardarán los modelos, y después los se cargarán:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_model1, 'xgboost_model1')
guardar_modelo(xgboost_model2, 'xgboost_model2')

y_pred1, model1 = cargar_validar_modelo('xgboost_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model2', features1, y1) #Contiene todos los datos, features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birth_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```


3.4.2.1.3.2. Técnica Regresión Logística

```
#REGRESIÓN LOGÍSTICA:
#El output es la unión de 2 modelos:
#1. 'BW_Percentile_inferior_10' - Muestra no estratificada
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10'.
print('\n*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
reg_logistica_model1 = get_regresion_logistica_model_individual_multiclass(features1, y1, y_column, muestra_estratificada)

#2. 'FGR_birth_nivel' - Muestra no estratificada
#Modelo 2: Predecir variable 'FGR_birth_nivel'.
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
dataset2 = dataset2[dataset2['FGR_birth_nivel']!=0]
dataset2['FGR_birth_nivel'] = dataset2['FGR_birth_nivel'].cat.remove_categories([0])
muestra_estratificada = True

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
reg_logistica_model2 = get_regresion_logistica_model_individual_multiclass(features2, y2, y_column, muestra_estratificada)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se cargan:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(reg_logistica_model1, 'reg_logistica_model1')
guardar_modelo(reg_logistica_model2, 'reg_logistica_model2')

y_pred1, model1 = cargar_validar_modelo('reg_logistica_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('reg_logistica_model2', features1, y1) #Contiene todos los datos, features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birth_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```

3.4.2.1.3.3. Técnica Red neuronal simple

```
#Construcción modelo de red neuronal:
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10'.
print('\n*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False
layers = best_parameters_rnn['layers']
lr = best_parameters_rnn['lr']
epochs = best_parameters_rnn['epochs']
batch_size = best_parameters_rnn['batch_size']
binaria = True

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1, y_column)
features1 = convertir_a_dummies(features1)
rnn_model1 = get_rnn_model_individual(features1, y1, y_column, layers, lr, epochs, batch_size, binaria)

#Construcción modelo de red neuronal:
#Modelo 2: Predecir variable 'BW_Percentile_inferior_10'.
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
dataset2 = dataset2[dataset2['FGR_birth_nivel']!=0]
dataset2['FGR_birth_nivel'] = dataset2['FGR_birth_nivel'].cat.remove_categories([0])
muestra_estratificada = False
layers = best_parameters_rnn['layers']
lr = best_parameters_rnn['lr']
epochs = best_parameters_rnn['epochs']
batch_size = best_parameters_rnn['batch_size']
binaria = False

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
y2 = y2.replace(1,0)
y2 = y2.replace(2,1)
y2 = y2.replace(3,2)

y2 = to_categorical(y2, num_classes=len(np.unique(y2)))

features2 = convertir_a_dummies(features2)
rnn_model2 = get_rnn_model_individual(features2, y2, y_column, layers, lr, epochs, batch_size, binaria)

#MODELO FINAL:
#No se incluirá este modelo debido a su baja predicción.
```

3.4.2.1.4. Estudio 4: Predicción de grado de FGR (Predicción variables FGR_birth_nivel)

En este estudio se intentará predecir si el neonato nació pequeño y su grado de FGR., con datos obtenidos dentro de los 2 primeros trimestres de embarazo., y en un único modelo.

3.4.2.1.4.1. Técnica XGBoost

```
#2. 'FGR_birth_nivel'
#Modelo 2: Predecir variable 'FGR_birth_nivel'.
print('\n*****Modelo 1*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = True
model_parameters = best_parameters_XGboost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
xgboost_model_2 = get_xgboost_model_individual(features, y, y_column, model_parameters)
```

3.4.2.1.4.2. Técnica Regresión Logística

```
#2. 'FGR_birth_nivel' - Muestra estratificada
#Modelo 2: Predecir variable 'FGR_birth_nivel'.
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = True

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
reg_logistica_model_2 = get_regresion_logistica_model_individual_multiclass(features, y, y_column, muestra_estratificada)
```

3.4.2.1.4.3. Técnica Red neuronal simple

```
#2. 'FGR_birth_nivel' - Muestra estratificada
#Modelo 2: Predecir variable 'FGR_birth_nivel'.
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = True
binaria = False
layers = best_parameters_rnn['layers']
lr = best_parameters_rnn['lr']
epochs = best_parameters_rnn['epochs']
batch_size = best_parameters_rnn['batch_size']

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
rnn_model_2 = get_rnn_model_individual(features, y, y_column, layers, lr, epochs, batch_size, binaria)
```

3.4.2.2. Predicción no temprana de problemas en el feto o nacimiento de neonato pequeño: Incluyendo también en el estudio las variables obtenidas en el tercer trimestre de gestación.

En el apartado 3.4.1., se explicaba que se durante el perfilado de variables, se había localizado un grupo de variables altamente explicativas, el cual correspondía a datos que no se obtenían durante los 2 primeros trimestres de gestación, y por este motivo no se utilizarían en el apartado de predicción temprana 3.4.2.1., sino que se utilizarían en este apartado 3.4.2.2. de predicción no temprana. Estas variables eran: **“Grupo de variables 'altamente explicativas’”**.

Si se analiza el apartado del anexo “Anexo 1: Estudio de importancia de variables por agrupación”, se podrá comprobar que estas variables corresponden a 2 pruebas médicas y que se habían almacenado en los dataframes “df_ultrasonido_fetal_3_trim”, “df_ultrasonido_doppler_1” y “df_ultrasonido_doppler_2” para su estudio. Con el fin de analizar la antelación con la que se hicieron las pruebas respecto a la fecha de parto, se realizará una comparación entre la fecha de parto de cada observación, y la fecha de realización de cada prueba:

```
data['Dias_entre_nacimiento_y_ultrasonido_periodico'] = (data['DateofDelivery']-data['Date_US_at_Diagnosis_Fetal_Suboptimal_Growth']).dt.days
data['Dias_entre_nacimiento_y_ultrasonido_periodico'].loc[data['Dias_entre_nacimiento_y_ultrasonido_periodico']<0] = 0
data['Dias_entre_nacimiento_y_ultrasonido_3trim'] = (data['DateofDelivery']-data['Date_US_at_Third_trim']).dt.days
data['Dias_entre_nacimiento_y_ultrasonido_3trim'].loc[data['Dias_entre_nacimiento_y_ultrasonido_3trim']<0] = 0
data['Dias_entre_nacimiento_y_ultimo_Doppler'] = (data['DateofDelivery']-data['Date_last_Doppler_US_before_delivery']).dt.days
data['Dias_entre_nacimiento_y_ultimo_Doppler'].loc[data['Dias_entre_nacimiento_y_ultimo_Doppler']<0] = 0
```

En el estudio se valorará la influencia de las mismas en la predicción, por lo que se determinará cuan de necesaria es la realización de estas pruebas. Se guardarán en una lista “vars_ultrasonido_periodico” las variables correspondientes a la prueba **Ecografía Doppler en 1ª visita del tercer trimestre**, del dataframe “df_ultrasonido_doppler_1”. Se guardarán en una lista “vars_ultrasonido_3trim” las variables correspondientes a la prueba **Ecografía fetal del tercer trimestre**, del dataframe “df_ultrasonido_fetal_3_trim” y se guardarán en una lista “vars_ultimo_doppler” las variables correspondientes a la prueba **Última Ecografía Doppler del tercer trimestre**, del dataframe “df_ultrasonido_doppler_2”:

```
#Variable 'Date_US_at_Diagnosis_Fetal_Suboptimal_Growth':
vars_ultrasonido_periodico = ['GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX', 'HC_DX', 'AC_DX', 'FL_DX', 'EFW_DX', 'PERCENTIL_EFW_DX', 'MVP_MM_DX', 'IP_AU_DX', 'IP_ACM_DX', 'IP_mAut_DX', 'Aortic_Ithsmus_DX', 'IP_DV_DX', 'AFI_DX']

#Existen 3 variables que contienen más del 40% de nulos, por lo que se deberán descartar del estudio:
descartar = ['Aortic_Ithsmus_DX', 'IP_DV_DX', 'MVP_MM_DX', 'AFI_DX']

vars_ultrasonido_periodico = ['GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX', 'HC_DX', 'AC_DX', 'FL_DX', 'EFW_DX', 'PERCENTIL_EFW_DX', 'IP_AU_DX', 'IP_ACM_DX', 'IP_mAut_DX']

#Variable 'Date_US_at_Third_trim':
vars_ultrasonido_3trim = ['GA_US_Third_trim', 'BPD_THIRDT', 'HC_THIRDT', 'AC_THIRDT', 'FL_THIRDT', 'EFW_THIRDT', 'PERCENTIL_EFW_THIRDT']

vars_ultimo_doppler = ['IP_AU_LastDoppler', 'IP_ACM_LastDoppler', 'Aortic_Ithsmus_LastDoppler', 'IP_DV_LastDoppler', 'Mat_angiogeni
```

De las 5 variables de 'vars_ultimo_doppler', existen 4 variables que contienen más del 40% de nulos, por lo que se deberán descartar del estudio. Además, la variable 'Mat_angiogenic', sólo indica si se realizó o no una prueba, por lo que también se descartará del estudio. De las 5 variables que se tenían de 'vars_ultimo_doppler' no se puede trabajar con ninguna, por lo que esto imposibilita realizar el estudio de la incidencia de esta prueba.

Se estudia la diferencia de fechas entre la obtención de las pruebas y la fecha de parto:

La media de la diferencia de días entre el nacimiento del neonato y la obtención de los datos de las variables ['GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX', 'HC_DX', 'AC_DX', 'FL_DX', 'EFW_DX', 'PERCENTIL_EFW_DX', 'IP_AU_DX', 'IP_ACM_DX', 'IP_mAut_DX'] son 42 días.

La media de la diferencia de días entre el nacimiento del neonato y la obtención de los datos de las variables ['GA_US_Third_trim', 'BPD_THIRDT', 'HC_THIRDT', 'AC_THIRDT', 'FL_THIRDT', 'EFW_THIRDT', 'PERCENTIL_EFW_THIRDT'] son 34 días.

La media de la diferencia de días entre el nacimiento del neonato y la obtención de los datos de las variables ['IP_AU_LastDoppler', 'IP_ACM_LastDoppler', 'Aortic_Ithsmus_LastDoppler', 'IP_DV_LastDoppler', 'Mat_angiogenic'] son 9 días.

De las 1054 observaciones no nulas de la variable 'Dias_entre_nacimiento_y_ultrasonido_periodico' (en total la variable contiene 132 nulos) 283 observaciones se obtuvieron más de 60 días antes del nacimiento, y 448 se obtuvieron dentro de los últimos 30 días antes del parto, con una media de 42 días.

De las 912 observaciones no nulas de la variable 'Dias_entre_nacimiento_y_ultrasonido_3trim' (en total la variable contiene 274 nulos) 73 observaciones se obtuvieron más de 60 días antes del nacimiento, y 422 se obtuvieron dentro de los últimos 30 días antes del parto, con una media de 34 días.

De las 675 observaciones no nulas de la variable 'Dias_entre_nacimiento_y_ultimo_Doppler' (en total la variable contiene 511 nulos) 17 observaciones se obtuvieron más de 60 días antes del nacimiento, y 637 se obtuvieron dentro de los últimos 30 días antes del parto, con una media de 9 días.

Teniendo en cuenta que la variable vars_ultrasonido_periodico contiene los fenotipos obtenidos de la ecografía Doppler con una media de 42 días de antelación y que la variable vars_ultrasonido_3trim contiene los fenotipos obtenidos de la ecografía fetal del tercer

trimestre con una media de 34 días de antelación, se utilizará esta nueva información para realizar los siguientes estudios, los cuales se aplicarán al modelo seleccionado construido en la sección 4.2., pero incorporando estas nuevas variables:

- a) **ENFOQUE 1:** Agregar al modelo seleccionado las variables 'vars_ultrasonido_periodico' (ecografía Doppler)
 - Estudio 1: Agregar variables al modelo
 - Perfilado de las variables del Estudio 1
- b) **ENFOQUE 2:** Agregar al modelo seleccionado las variables perfiladas del enfoque 1 y las variables de 'vars_ultrasonido_3trim' (ecografía fetal del tercer trimestre)
 - Estudio 2: Agregar variables al modelo
 - Perfilado de las variables del Estudio 2

Se decide no utilizar las variables contenidas en vars_ultimo_doppler correspondientes a la última ecografía Doppler realizada, ya que la media de obtención de los datos es de 9 días de antelación al parto y además contiene una gran cantidad de valores nulos, aunque, como se ha comentado, 9 días son oro para tratar de aplicar tratamientos al feto para, por ejemplo, madurarlo lo máximo posible antes de que la madre dé a luz.

3.4.2.2.1. Estudio 1: Agregar al modelo las variables de la ecografía Doppler

Se agregarán las variables 'vars_ultrasonido_periodico' al modelo que sea seleccionado del apartado 4.2., para comprobar si la precisión del modelo mejora al agregar estas variables obtenidas en el tercer trimestre de embarazo.

3.4.2.2.2. Estudio 2: Agregar al modelo las variables perfiladas del estudio 1 y las variables de la ecografía fetal del tercer trimestre

Se agregarán las variables 'vars_ultrasonido_3trim' al modelo que sea seleccionado del apartado 4.2., junto con las variables perfiladas 'vars_ultrasonido_periodico' para comprobar si la precisión del modelo mejora al agregar estas variables obtenidas en el tercer trimestre de embarazo.

4. Experimentos, validación y resultados

4.1. Introducción

Para evaluar los modelos se deciden utilizar las siguientes métricas:

- **accuracy_score:** Devolverá la precisión del modelo en función del número de aciertos del mismo, es decir, valorará cuan de cerca está el modelo de hacer una clasificación correcta.
- **Matriz de confusión:** Mostrará el número de verdaderos positivos y negativos, y el número de falsos positivos y negativos, es decir, indicará el número de aciertos del modelo en cada categoría, y mostrará el número de los que no acertó y su categoría clasificada. Los valores que devuelve son:

- Verdaderos positivos: Número de positivos que el modelo clasificó correctamente como positivos.
 - Verdaderos negativos: Número de negativos que el modelo clasificó correctamente como negativos.
 - Falsos negativos: Número de positivos que el modelo clasificó incorrectamente como negativos.
 - Falsos positivos: Número de negativos que el modelo clasificó incorrectamente como positivos.
- **Reporte de clasificación**: Mostrará las siguientes métricas:
 - Precision: Cuánto acierta el modelo.
 - Recall: Ofrece el porcentaje de clasificaciones de cada categoría.
 - f1-score: Combina la precisión (accuracy) y la sensibilidad (recall) en una única métrica, y es de gran utilidad cuando la distribución de la muestra no es equilibrada, es decir, que el número de observaciones por clase es muy diferente. Se calcula con la siguiente fórmula: $2 \times (\text{Recall} \times \text{Precisión}) / (\text{Recall} + \text{Precisión})$ y ofrece la siguiente información:
 - Altos porcentajes de Accuracy y Recall: El modelo clasifica bien la clase.
 - Alto porcentaje de Accuracy y bajo de Recall: El modelo no clasifica demasiado bien la clase, pero cuando lo hace es altamente confiable.
 - Bajo porcentaje de Accuracy y alto de Recall: El modelo clasifica bien la clase pero también incluye clasificaciones de otras clases.
 - Bajos porcentajes de Accuracy y Recall: El modelo no clasifica bien la clase.
 - Support: Indica el número de observaciones o clasificaciones de cada categoría en el set de test.
 - **Curva ROC**: Curva que muestra el rendimiento de un modelo de clasificación en todas las entradas de clasificación. La curva representa 2 parámetros: Tasa de verdaderos positivos (tpr) y tasa de falsos positivos (fpr).
 - **AUC (Área bajo la curva ROC)**: Representa la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. El AUC es invariable con respecto al umbral de clasificación, así como a la escala, ya que mide qué tan bien se clasifican las predicciones, en lugar de sus valores absolutos.
 - **Coefficiente gini**: Es el resumen estadístico de la tabla de perfil de precisión acumulativa (CAP). Se calcula como el cociente del área que encierra la curva CAP y la diagonal y el área correspondiente en un procedimiento de calificación ideal. Su fórmula es $(2 \times \text{AUC}) - 1$.

4.2. Predicción temprana de problemas en el feto o nacimiento de neonato pequeño: Mediante el uso de variables que se puedan obtener dentro de los 2 primeros trimestres de embarazo.

En este apartado se comentarán los resultados obtenidos con las variables contenidas en el dataframe “variables_seleccionadas_estudio_FGR” en cada arquitectura del apartado 4.2.1:

4.2.1. Estudio 1: Predicción del percentil de crecimiento en el nacimiento (Predicción de la variable BW_Percentile)

4.2.1.1. Técnica XGBoost

Con las dos opciones de eliminar o modificar los datos nulos, se obtiene una precisión muy baja (12% en cada uno), mucho menor que si se escogiera la clasificación de manera aleatoria (50%).

El nivel de estudio al que se está intentando llegar es muy profundo y concreto, existen 100 clasificaciones diferentes, por lo que se considera difícil realizar esta predicción y poder predecir el percentil de nacimiento del neonato.

Aunque el valor de AUC pueda confundir, se puede ver que el gráfico generado es extraño, y esto es debido a que existen variables que tienen mucha influencia a la hora de clasificar, por ejemplo 'PE' realiza la primera división en el árbol del primer modelo y en el segundo 'Steroids_during_gestation', o 'EFW_second' tiene gran importancia en el primero y 'Maternal_pre_gestational_weight' en el segundo.

Si se analizan los valores del reporte de clasificación del segundo modelo, se puede ver que únicamente predice con un 26% de precisión el percentil 0 (existen 36 observaciones con percentil 0 (columna support), y el modelo sólo clasifica correctamente el 26%, es decir, 9) lo que hace que la precisión media del modelo aumente, sin embargo, este modelo no acierta prácticamente ningún percentil más, el siguiente que más acierta es el percentil 6 con un 17% de precisión.

Como el objetivo de estudio es detectar cuándo un neonato nacerá con problemas (percentil de nacimiento inferior a 10), se planteará el estudio desde otra perspectiva. Primero se intentará predecir si el neonato nacerá con problemas (percentil inferior a 10) para lo se binarizará la variable 'BV_Percentil' siendo 0 para los neonatos con percentil igual o superior a 10, y 1 para el resto. Posteriormente, para intentar realizar una clasificación dentro de los neonatos nacidos con problemas de salud, se crearán las categorías 1 (FGR Leve), 2 (FGR Moderado) y 3 (FGR Grave) para neonatos nacidos con percentil entre 5 y 10, entre 2 y 5 e inferior a 2 respectivamente.

4.2.2. Estudio 2: Predicción de feto con problemas (nacimiento de neonato de variable 'BW_Percentile_inferior_10)

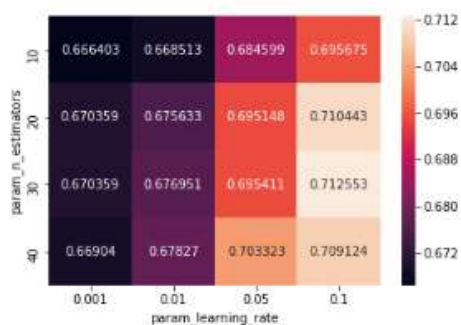
4.2.2.1. Técnica XGBoost

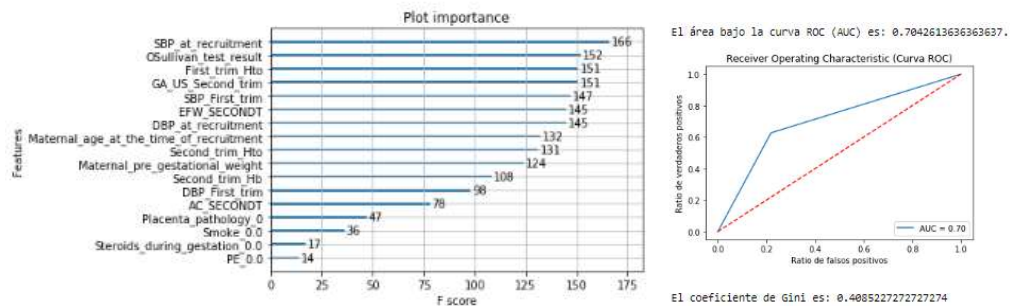
4.2.2.1.1. Búsqueda mejores hiperparámetros GridSearCV() y modelo con observaciones con nulos corregidas

Los parámetros que dan mejores resultados son:
{'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 30}

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.71	0.78	0.74	128
1	0.71	0.63	0.67	110
accuracy			0.71	238
macro avg	0.71	0.70	0.71	238
weighted avg	0.71	0.71	0.71	238





Se puede ver que los mejores parámetros que ofrece gridSearchCV() son 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 30, mediante los cuales se obtiene mejor precisión media de test. Asimismo, se puede ver el resto de mejores precisiones en el heatmap, donde se muestra una matriz de 4 x 4.

Si se analiza la matriz de confusión, se comprobará que el modelo seleccionado consigue clasificar correctamente 100 de 128 categorías de clase 0 y 69 de 110 de clase 1, ofreciendo una precisión total de 71%, un AUC de 70,42 %, lo cual representa la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio, lo cual nos interesa especialmente para este estudio, ya que se trata de diagnósticos médicos y siempre será menos grave la clasificación de un falso positivo que la de un falso negativo.

Asimismo, se puede ver en el reporte de clasificación que existen valores altos de precisión y recall para la clase 0, y no tan altos en recall para la clase 1, por lo que se puede suponer que en la clase 0 el modelo clasifica bien la clase, pero no acierta tanto en la clase 1, aunque los aciertos que realiza son confiables. Estas dos métricas de precisión y recall se combinan en la métrica f1-score que, como se ha explicado, es el resultado de aplicar la fórmula $2 \times (\text{Recall} \times \text{Precisión})$.

También se puede comprobar que la forma de la curva ROC es muy particular, hace un pico. Esto significa que el modelo está muy influenciado por pocas variables, es decir, el árbol se divide muy fuertemente según unas variables, es decir, estas explican mucha varianza.

Asimismo, se puede comprobar que el modelo ofrece un buen AUC, lo cual, como se ha comentado, es especialmente positivo para este proyecto.

El valor de coeficiente gini es de 40%, ya que depende del valor de AUC.

NOTA: En los siguientes apartados la explicación no será tan densa como esta para evitar alargar la memoria, ya que hay un gran número de apartados. Como al inicio de esta sección se ha explicado detalladamente el significado de cada métrica, se tomarán estos conceptos como referencia y en los siguientes apartados no se detallará tanto su significado.

4.2.2.1.2. Eliminar observaciones con nulos

Este modelo es igual al anterior, ya que toma los parámetros del mejor modelo obtenido, pero los datos de entrada no serán los mismos, ya que se eliminan las observaciones y en el anterior se modificaban. Se puede ver que en este modelo la precisión y el AUC mejoran en un 3%, pero se ha contado con 487 observaciones menos y, por lo tanto, el conjunto de test se ha reducido de 238 a 140 y el modelo no tiene tantos datos que clasificar. Concretamente, en el primero modelo se clasifican correctamente 169 casos y en este 104. Entre una opción y otra la más adecuada sería la primera.



Reporte de clasificación:					
	precision	recall	f1-score	support	
0	0.76	0.80	0.78	79	
1	0.72	0.67	0.69	61	
accuracy			0.74	140	
macro avg	0.74	0.73	0.74	140	
weighted avg	0.74	0.74	0.74	140	

4.2.2.2. Técnica Regresión logística

4.2.2.2.1. Eliminar observaciones con nulos

Se puede comprobar que la precisión del modelo mejora aproximadamente un 1,5% respecto al anterior y un 05% el AUC, bajo las mismas condiciones de eliminación de datos, y clasifica mucho mejor la clase 0 que la 1.

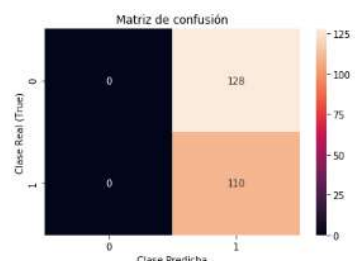
4.2.2.2.2. Corregir observaciones con nulos

En el caso de no eliminar las observaciones, sino modificarlas, la precisión del modelo empeora respecto a la opción de eliminarlas o en cualquiera de las opciones de XGBoost, aunque mantiene la tendencia de los anteriores modelos de clasificar mejor la clase 0 (neonato no nacido pequeño) que la 1 (neonato pequeño).

4.2.2.3. Técnica Red neuronal simple

4.2.2.3.1. Búsqueda mejores hiperparámetros GridSearCV() y modelo con observaciones con nulos corregidas

Como se indicó en el apartado anterior, no se obtenían buenos resultados con ningún conjunto de parámetros:



	mean_test_score	rank_test_score	param_epochs	param_layers	param_lr
0	0.488397	1	10	[256]	0.1
15	0.488397	1	50	[128, 128]	0.1
14	0.488397	1	50	[256]	0.001
13	0.488397	1	50	[256]	0.01
12	0.488397	1	50	[256]	0.1

Se puede comprobar que este modelo de red no predice en ningún caso la clase 0, y la precisión falsa del modelo de 46% es debido a que acierta todos los casos en que el modelo no es 0, ya que únicamente predice la clase 0, aun habiendo probado con muchos parámetros y arquitectura de red diferentes, tal y como se ha mencionado, siempre se llegaba a la misma calidad de predicción.

4.2.2.3.2. Eliminar observaciones con nulos

Se puede comprobar que eliminando las observaciones tampoco cambia el resultado, por lo que se puede determinar que la red neuronal no funciona para este problema de clasificación de predicción temprana de nacimiento de neonato pequeño. Sin embargo, se seguirá construyendo la red neuronal para el resto de estudios para ver si la red consigue aprender.

4.2.3. Estudio 3: Predicción de feto con problemas y grado de FGR (Predicción variables BW_Percentile_inferior_10 y FGR_birth_nivel)

4.2.3.1. Técnica XGBoost

Debido a que se pierden muchas observaciones al eliminar los valores nulos y, como se verá en el desarrollo, la precisión del modelo no es mejor que asignando la mediana a las variables numéricas y una categoría aleatoria de la variable a las variables categóricas, no se hará la concatenación de modelos en los casos de eliminación de observaciones. Sólo se aplicará la unión de los dos modelos cuando no se eliminen más del 30% de las observaciones.

4.2.3.1.1. Eliminar observaciones con nulos

Analizando el modelo por separado, es decir, modelo a modelo, se puede comprobar que la precisión del modelo 1 es de 74,28%, igual que en la sección anterior, ya que el estudio es el mismo.

Analizando el modelo 2, donde se han separado los casos 0 para que al estratificar la muestra esta se reduzca lo menos posible, se puede comprobar que el modelo clasifica correctamente 10 de 17 casos la clase 1, 8 de 14 la clase 2 y 11 de 30 la clase 3, es decir, y clasifica mejor la clase 1 (mayor recall), es decir, cuando el grado de FGR es leve (percentil entre 5 y 10).

4.2.3.1.2. Corregir observaciones con nulos

Analizando el modelo por separado, es decir, modelo a modelo, se puede comprobar que la precisión del modelo 1 es de 71%, igual que en la sección anterior, ya que el estudio es el mismo.

Analizando el modelo 2, donde se han separado los casos 0 para que al estratificar la muestra esta se reduzca lo menos posible, se puede comprobar que se dispone de 115 observaciones para test y 312 para train (menor número de muestras para entrenar, 104 por clase) el modelo clasifica correctamente 18 de 33 casos la clase 1, 6 de 22 la clase 2 y 33 de 60 la clase 3, es decir, y clasifica mejor la clase 3 (mismo recall que en la clase 1, pero muestra el doble de grande), es decir, cuando el grado de FGR es severo (percentil inferior a 2). Asimismo, se obtiene un AUC del 72% y un gini de 43,63%.

Pero lo más interesante es analizar el verdadero output que es la unión de los dos modelos, donde se verá realmente cómo responde este modelo a datos nuevos que nunca ha visto (recordar que el modelo sólo se entrenó con 312 observaciones de las 1186 del set, y ahora se testeará contra esas 1186).

Se puede comprobar que la precisión de este modelo final es de 80,43%, muy alta teniendo en cuenta que únicamente se cuenta con 17 variables, de las cuales tan sólo 5 se obtuvieron de pruebas médicas de coste, el resto procedían de cuestionarios o fueron pruebas para medir la tensión. Asimismo, mencionar que únicamente se dispone de dos variables del ultrasonido del primer trimestre 'CRL' y 'BPD_First', y de estas dos únicamente se pudo utilizar 'CRL' debido a la gran cantidad de nulos de 'BPD_First', y 'CRL' fue descartada por no ser explicativa. El modelo consigue clasificar correctamente 581 de 613 clases 0, es decir, clasifique como clasifique el resto lo más importante es que el modelo indica que todas las demás clasificaciones son predicciones de neonato nacido con restricción, por lo que se considera un buen modelo de detección temprana. Del resto de clasificaciones, consigue clasificar correctamente 106 de 161 de la clase 1, 99 de 126 de la clase 2 y 168 de 286 de la clase 3.

4.2.3.2. Técnica Regresión logística

4.2.3.2.1. Eliminar observaciones con nulos

Analizando el modelo por separado, es decir, modelo a modelo, se puede comprobar que la precisión del modelo 1 es de 75,71%, igual que en la sección anterior, ya que el estudio es el mismo.

Analizando el modelo 2, donde se han separado los casos 0 para que al estratificar la muestra esta se reduzca lo menos posible, se puede comprobar que el modelo clasifica correctamente 12 de 17 casos la clase 1, 0 de 14 la clase 2 y 16 de 30 la clase 3, es decir, y clasifica mejor la clase 1 (mayor recall), es decir, cuando el grado de FGR es leve (percentil entre 5 y 10).

4.2.3.2.2. Corregir observaciones con nulos

Analizando el modelo por separado, es decir, modelo a modelo, se puede comprobar que la precisión del modelo 1 es de 67,65%, igual que en la sección anterior, ya que el estudio es el mismo.

Analizando el modelo 2, donde se han separado los casos 0 para que al estratificar la muestra esta se reduzca lo menos posible, se puede comprobar que el modelo clasifica correctamente 20 de 33 casos la clase 1, 5 de 22 la clase 2 y 34 de 60 la clase 3, es decir, y clasifica mejor la clase 3 (mejor f1-score), es decir, cuando el grado de FGR es severo (percentil inferior a 2). Asimismo, se obtiene un AUC del 71% y un gini de 41,69%.

En relación al output final, consecuencia de la unión de los dos modelos, se puede comprobar que la precisión de este modelo final es de 58,43%. El modelo consigue clasificar correctamente 489 de 613 clases 0, es decir, falla en 124 casos indicando una predicción positiva cuando en realidad es negativa.

Del resto de clasificaciones, consigue clasificar correctamente 28 de 161 de la clase 1, 24 de 126 de la clase 2 y 152 de 286 de la clase 3.

En definitiva, hay un 30% de diferencia de precisión entre el modelo construido con la técnica XGBoost y este con regresión logística, con los mismos datos de entrada y transformaciones, por lo que de momento se escogería el modelo del apartado **4.2.3.1.2.**

4.2.3.3. Técnica Red neuronal simple

4.2.3.3.1. Eliminar observaciones con nulos

Como se explicaba en apartados anteriores, la red neuronal no está siendo una buena técnica para tratar este problema. En este caso, únicamente consigue predecir la clase 1 en el primer modelo, y la clase 2 (equivalente a la clase 3) en el segundo.

4.2.3.3.2. Corregir observaciones con nulos

En este modelo sucede lo mismo que en el anterior. Por este motivo, se decide no guardar los modelos ni construir el modelo final debido a su baja predicción.

4.2.4. Estudio 4: Predicción de grado de FGR (Predicción variables FGR_birth_nivel)

En este apartado el objetivo es el mismo que en el apartado anterior, predecir la variable **FGR_birth_nivel**, con la diferencia de que en este estudio no se separará la clase 0 de las demás, se intentarán predecir directamente todas las categorías juntas. El inconveniente de realizar esto es que al estratificar la muestra se perderán muchas variables con las que entrenar, pero no para testear, ya que para construir el modelo lo más robusto posible para evitar el overfitting, aunque se entrene con

menos muestras (concretamente, el número de observaciones de la menor clase multiplicado por el número de clases diferentes del dataset de entrada), se testeará contra todo el conjunto de test.

Se compararán los resultados de este estudio con los obtenidos en el estudio 3, para analizar qué modelo ofrece mejores resultados de predicción.

4.2.4.1. Técnica XGBoost

4.2.4.1.1. Eliminar observaciones con nulos

Se puede comprobar que la precisión obtenida es de 50,71%, una precisión muy baja, especialmente si se compara con la precisión obtenida mediante la aplicación de los dos modelos del apartado anterior mediante la misma técnica.

4.2.4.1.2. Corregir observaciones con nulos

En este caso la predicción empeora mucho, siendo la precisión del 43,69% y con valores muy negativos en el reporte de clasificación. Con respecto al modelo de la sección anterior, se está rebajando la precisión en aproximadamente un 40% con respecto a la construcción de un modelo a partir de dos.

4.2.4.2. Técnica Regresión logística

4.2.4.2.1. Eliminar observaciones con nulos

Se comprueba que la precisión obtenida es de 50%, una precisión muy baja, especialmente si se compara con la precisión obtenida mediante la aplicación de los dos modelos del apartado anterior mediante la misma técnica.

4.2.4.2.2. Corregir observaciones con nulos

En este caso la predicción mejora un 1,26% respecto al anterior, siendo la precisión del 51,26% y con valores muy negativos en el reporte de clasificación. Con respecto al modelo de la sección anterior, se reduce la precisión en aproximadamente un 7% con respecto a la construcción de un modelo a partir de dos.

4.2.4.3. Técnica Red neuronal simple

4.2.4.3.1. Eliminar observaciones con nulos

En este caso, la red neuronal es capaz de predecir las 4 clases, aunque sólo acertando en 3 de ellas. Respecto a la clase 0, predice 60 de 79 casos, falla en 19, lo que mejora mucho los resultados anteriores obtenidos con la red. Predice 8 de 19 de la clase 1, 0 de 15 de la clase 2 y 11 de 27 de la clase 3. De momento, este es el modelo de red neuronal que mejores resultados ha dado, consigue predecir mejor las clases, pero sigue siendo un modelo no adecuado para el estudio por sus deficientes resultados.

La precisión del modelo es de 56,42 %.

4.2.4.3.2. Corregir observaciones con nulos

En este caso el resultado empeora mucho, la precisión del modelo es 23,94% ya que acierta 38 de 39 clasificaciones de la clase 1, pero el acierto de las demás clases es muy bajo o es nulo. Definitivamente la red neuronal no es una buena técnica para el set de datos ni para el objetivo del proyecto.

4.3. Predicción no temprana de problemas en el feto o nacimiento de neonato pequeño: Incluyendo también en el estudio las variables obtenidas en el tercer trimestre de gestación.

En el apartado 3.4.1., se estuvieron buscando los fenotipos para predecir de manera temprana (dentro de los 2 primeros trimestres de gestación) si el feto presentaba problemas y su grado de restricción, y después de aplicar diferentes estudios y diferentes técnicas, se seleccionaron, de entre todos los modelos construidos, el que mejores resultados ofrecía. Este modelo se muestra en el apartado 4.5.1.

En este apartado se realizará un enfoque diferente para tratar de obtener un mejor modelo de predicción, utilizando las variables que se obtienen de las pruebas médicas realizadas en el tercer trimestre. Para ello será necesario valorar si es sostenible utilizar estas variables para construir el modelo, ya que, si las pruebas se realizan muy próximas al parto, tal vez no habrá tiempo de reacción, aunque se debe tener en cuenta que detectar que un neonato nacerá antes de tiempo con una semana de antelación, ya es más que no detectarlo hasta los dos últimos días, ya que se pueden aplicar tratamientos médicos, por ejemplo, para madurar los órganos en la medida de lo posible.

4.3.1. Estudio 1: Agregar al modelo las variables de la ecografía Doppler

4.3.1.1. Agregar variables al modelo

En primer lugar, se añadirán las variables “vars_ultrasonido_periodico” al modelo seleccionado:

```
##Modelo 1: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,data[vars_ultrasonido_periodico],y],axis=1)
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
xgboost_model_vars_periodico1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

##Modelo 2: Predecir variable 'FGR_birth_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
#esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
#la clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,data[vars_ultrasonido_periodico],y],axis=1)
dataset2 = dataset2[dataset2['FGR_birth_nivel']!=0]
dataset2['FGR_birth_nivel'] = dataset2['FGR_birth_nivel'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
xgboost_model_vars_periodico2 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

##MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
cargarán:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_model_vars_periodico1, 'xgboost_model_vars_periodico1')
guardar_modelo(xgboost_model_vars_periodico2, 'xgboost_model_vars_periodico2')

y_pred1, model1 = cargar_validar_modelo('xgboost_model_vars_periodico1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model_vars_periodico2', features1, y1) #Contiene todos los datos,
features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birth_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```


En comparación con el modelo seleccionado para dar respuesta al objetivo principal del proyecto de predicción temprana de FGR, al agregar al modelo las variables de la ecografía Doppler contenidas en `vars_ultrasonido_periodico`, se comprueba que la precisión de este pasa de ser del 80,43 % a 87,27%. Una mejora sorprendente, logrando clasificar correctamente el 96% de la clase 0 (con esta información ya se sabía que el resto de fetos presentarán problemas), concretamente 589 de 613 de las clasificaciones de la clase 0, 124 de 161 de la clase 1, 103 de 126 de la clase 2 y 219 de 286 de la clase 3, siendo la clase peor predicha la 2, correspondiente a FGR Moderado.

4.3.1.2. Perfilado de las variables del Estudio 1

A continuación, se intentará mejorar el modelo mediante el análisis de las métricas y datos del estudio del apartado anterior, y se tratará de reducir el número de variables actual que ha pasado a ser de 35. Se comenzará analizando los árboles y los gráficos de importancia de variables del punto anterior y se realizarán diferentes combinaciones tratando de obtener los mejores resultados.

Después de un largo proceso de perfilado, se concluye que las variables que mejoran el modelo son las siguientes, los cuales se guardarán en la variable `utilizar_periodico` para utilizarla en el siguiente apartado:

`utilizar_periodico = 'IP_ACM_DX', 'PERCENTIL_EFW_DX', 'IP_mAut_DX', 'SBP_at_recruitment', 'Maternal_age_at_the_time_of_recruitment', 'OSullivan_test_result', 'Maternal_pre_gestational_weight', 'EFW_SECOND', 'First_trim_Hto', 'GA_US_Second_trim', 'IP_AU_DX', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX'`

```
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
utilizar_periodico = ['IP_ACM_DX', 'PERCENTIL_EFW_DX', 'IP_mAut_DX', 'SBP_at_recruitment', 'Maternal_age_at_the_time_of_recruitment', 'OSullivan_test_result', 'Maternal_pre_gestational_weight', 'EFW_SECOND', 'First_trim_Hto', 'GA_US_Second_trim', 'IP_AU_DX', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX']
dataset1 = pd.concat([data[utilizar_periodico], y], axis=1)
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}
features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1, y_column)
features1 = convertir_a_dummies(features1)
xgboost_model_vars_periodico1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'FGR_birth_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de la clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birth_nivel'
y = data[y_column]
dataset2 = pd.concat([data[utilizar_periodico], y], axis=1)

dataset2 = dataset2[dataset2['FGR_birth_nivel'] != 0]
dataset2['FGR_birth_nivel'] = dataset2['FGR_birth_nivel'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}
features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2, y_column)
features2 = convertir_a_dummies(features2)
xgboost_model_vars_periodico2 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se cargarán:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_model_vars_periodico1, 'xgboost_model_vars_periodico1')
guardar_modelo(xgboost_model_vars_periodico2, 'xgboost_model_vars_periodico2')

y_pred1, model1 = cargar_validar_modelo('xgboost_model_vars_periodico1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model_vars_periodico2', features1, y1) #Contiene todos los datos, features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birth_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```

Después de la fase de perfilado, se consigue obtener una mejora en la precisión del modelo, concretamente un 88,19 %, con tan sólo 13 variables, logrando así clasificar correctamente 593 de 613 en la clase 0, 126 de 161 de la clase 1, 102 de 126 de la clase 2 y 225 de 286 de la clase 3, consiguiendo así no sólo mejorar las clasificaciones de la clase 0 que permitirán identificar los fetos sanos, sino también mejora el poder de predicción del resto de clases de los niveles de FGR, lo cual permite alcanzar el objetivo secundario del proyecto que era la predicción no temprana de problemas en el feto y grado de FGR.

Se guardan los modelos por separado con el nombre '**xgboost_model_vars_periodico1**' y '**xgboost_model_vars_periodico2**', los cuales se unificarán en uno siguiendo los pasos del apartado 4.5. y se convierte en el modelo seleccionado para predecir el grado de FGR con los datos de la ecografía Doppler.

4.3.2. Estudio 2: Agregar al modelo las variables perfiladas del estudio 1 y las variables de la ecografía fetal del tercer trimestre

4.3.2.1. Agregar variables al modelo

A continuación, se parte de las 13 variables del modelo anterior y se añaden al estudio las variables de la ecografía del tercer trimestre, contenidas en “**vars_ultrasonido_3trim**”:

```
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no:
print('*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([data[utilizar_periodico],data[vars_ultrasonido_3trim],y],axis=1)
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
xgboost_model_vars_periodico_3trim1_1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'FGR_birthing_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
#esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
#La clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birthing_nivel'
y = data[y_column]
dataset2 = pd.concat([data[utilizar_periodico],data[vars_ultrasonido_3trim],y],axis=1)
dataset2 = dataset2[dataset2['FGR_birthing_nivel']!=0]
dataset2['FGR_birthing_nivel'] = dataset2['FGR_birthing_nivel'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
xgboost_model_vars_periodico_3trim2_1 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
cargarán:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_model_vars_periodico_3trim1_1, 'xgboost_model_vars_periodico_3trim1_1')
guardar_modelo(xgboost_model_vars_periodico_3trim2_1, 'xgboost_model_vars_periodico_3trim2_1')

y_pred1, model1 = cargar_validar_modelo('xgboost_model_vars_periodico_3trim1_1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model_vars_periodico_3trim2_1', features1, y1) #Contiene todos
los datos, features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birthing_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```

Se comprueba que al añadir las 7 variables del ultrasonido del tercer trimestre la precisión se reduce en un 0,42% respecto al modelo construido en el apartado anterior. Comparando las clasificaciones, se comprueba que existe una mejoría en la clasificación de las clases 0, 1 y 2, pero se reduce en la de la clase 3, concretamente, se pasa de clasificar correctamente 1046 casos a 1041, de ahí la reducción en la precisión.

4.3.2.2. Perfilado de las variables del Estudio 2

```
#Modelo 1: Predecir variable 'BW_Percentile_inferior_10', es decir, si el neonato nace pequeño o no;
print('*****Modelo 1*****')

#Parámetros:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]

utilizar_periodico_3trim = ['PERCENTIL_EFW_DX', 'IP_ACM_DX', 'IP_Aut_DX', 'SBP_at_recruitment', 'First_trim_Hto',
'IP_AU_DX', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'Maternal_age_at_the_time_of_recruitment',
'O'Sullivan_test_result', 'GA_US_Second_trim', 'PERCENTIL_EFW_THIRD', 'Maternal_pre_gestational_weight',
'BPD_THIRD', 'EFW_SECOND', 'BPD_DX']

dataset1 = pd.concat([data[utilizar_periodico_3trim], y], axis=1)
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1, y_column)
features1 = convertir_a_dummies(features1)
xgboost_model_vars_periodico_3trim1_2 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'FGR_birthing_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
#esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
#la clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'FGR_birthing_nivel'
y = data[y_column]
dataset2 = pd.concat([data[utilizar_periodico_3trim], y], axis=1)

dataset2 = dataset2[dataset2['FGR_birthing_nivel'] != 0]
dataset2['FGR_birthing_nivel'] = dataset2['FGR_birthing_nivel'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2, y_column)
features2 = convertir_a_dummies(features2)
xgboost_model_vars_periodico_3trim2_2 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
#guardan:
print('\n*****Modelo final = Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_model_vars_periodico_3trim1_2, 'xgboost_model_vars_periodico_3trim1_2')
guardar_modelo(xgboost_model_vars_periodico_3trim2_2, 'xgboost_model_vars_periodico_3trim2_2')

y_pred1, model1 = cargar_validar_modelo('xgboost_model_vars_periodico_3trim1_2', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model_vars_periodico_3trim2_2', features1, y1) #Contiene todos los datos, features2 e y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birthing_nivel']

reportes_modelo_cargado(y_test, y_pred_final)
```

Después de la fase de perfilado, se consigue obtener una mejora en la precisión del modelo respecto al del apartado 9.1.2.1., consiguiendo igualar la precisión del modelo seleccionado en la sección 9.1.1.2., concretamente un 88,19 %, con tan sólo 15 variables, 2 más que en el modelo seleccionado con las variables de la ecografía Doppler, logrando así conseguir clasificar el mismo número de muestras, pero en clases diferentes, por ejemplo, este modelo mejora la clasificación de las clases 0 y 2, pero empeora la clasificación de las clases 1 y 3, aunque el número de aciertos es el mismo.

Se guardan los modelos por separado con el nombre 'xgboost_model_vars_periodico_3trim1_2' y 'xgboost_model_vars_periodico_3trim2_2', los cuales se unificarán en uno siguiendo los pasos del apartado 8.2.5.1. y se convierte en el modelo seleccionado para predecir el grado de FGR con los datos de la ecografía fetal del tercer trimestre.

4.4. Procedimiento para unir 3 modelos en uno único (output)

Los métodos a utilizar son los siguientes:

```

#Función que obtiene la predicción entre 2 modelos:
def get_y_pred_modelo_final(y_pred1, y_pred2):
    y_pred1, y_pred2
    y_pred = y_pred1 * y_pred2
    return(y_pred)

#Función que guarda el modelo creado en la unidad por defecto:
def guardar_modelo(model, name_model):
    pickle.dump(model, open(name_model, 'wb'))

#Función para cargar el modelo guardado en la unidad por defecto:
def cargar_validar_modelo(model, features, y_test):
    model = joblib.load(model)
    y_pred = model.predict(features)
    return(y_pred, model)

#Función que imprime los reportes del modelo:
def reportes_modelo_cargado(y_test, y_pred):
    print('\nLa precisión del modelo es: {}'.format(accuracy_score(y_test, y_pred)))
    if(len(np.unique(y_pred))<7):
        print('\nMatriz de confusión: {}'.format(matriz_confusion(y_test, y_pred)))
    AUC = get_AUC(data['FGR_birth_nivel'], y_pred)
    get_gini(AUC)

```

Los pasos a seguir para poder unir los dos modelos y probarlo con datos nuevos es el siguiente:

1. Guardar cada modelo en un fichero: Para ello se llama al método construido **guardar_modelo()** y se le pasará el modelo que se desea guardar y el nombre con el que se quiere guardar.
2. Cargar cada modelo llamando a la función **cargar_validar_modelo()**, pasando como parámetro el nombre del fichero que contiene el modelo, el set de datos sin la variable objetivo que se desea pasar y la variable objetivo para poder hacer la predicción. Este método devolverá la predicción, es decir, la clase predicha por el modelo para cada observación, y el propio modelo.
3. Para unificar las dos predicciones se construye el método **get_y_pred_modelo_final()** al que se le pasará la predicción de los dos primeros modelos, teniendo en cuenta que el primero sólo predecirá la clase 0 y 1 y el segundo las clases 1, 2 y 3. De este modo, el método **get_y_pred_modelo_final()** ha sido preparado para esto teniendo en cuenta que cuando la predicción del primer modelo (**y_pred**) sea 0, la predicción del modelo concatenado (**y_pred_final**) será 0, y en caso de que sea 1, tomará el valor de la predicción del modelo 2 (**y_pred2**) para esa observación. Para conseguir esto se aplica la operación **y_pred_final = y_pred1 * y_pred2**.

4.5. Procedimiento para pasar nuevos datos al modelo (output)

Como se ha podido comprobar en el apartado 8.2.3.1.2., sección que ha dado lugar al modelo seleccionado, a los datos de entrada del modelo se le han aplicado las siguientes transformaciones para corregir los nulos:

- Corrección de los nulos de tipo numérico: Aplicar la mediana
- Corrección de los nulos de las variables categóricas: Escoger aleatoriamente una categoría de entre las de la clase.

Bien, teniendo esta información, se explicarán los pasos para poder cargar el modelo guardado y pasarle nuevos datos para realizar predicciones:

1. Aplicar a los datos de entrada las transformaciones mencionadas: Aplicar la mediana a los nulos de variables numéricas y un valor aleatorio de una de las categorías de las de la variable con nulos para las variables de tipo categórico.
2. Cargar cada modelo llamando a la función **cargar_validar_modelo()** para cada uno, pasando como parámetro el nombre del fichero que contiene el modelo, el set de datos sin la variable objetivo que se desean pasar y la variable objetivo para poder hacer la

- predicción. Este método devolverá la predicción, es decir, la clase predicha por el modelo para cada observación, y el propio modelo.
- Obtener la predicción conjunta de los dos modelos llamando al método **get_y_pred_modelo_final()**, pasando por parámetro el nombre del fichero que contiene el modelo, el nuevo set de datos separado de la variable objetivo, y la variable objetivo.
 - Llamar a la función **reportes_modelo_cargado()** pasando como parámetro la variable objetivo y la predicción concatenada. Esto mostrará diferentes métricas del modelo final.

4.6. Resumen de Fenotipos y modelos seleccionados

4.6.1. Modelo 1: Predicción temprana de feto con problemas y grado de FGR (Unión de 2 modelos)

4.6.1.1. Selección del modelo

El modelo seleccionado por sus buenos resultados de predicción (precisión de 80,43% prediciendo si un neonato nacerá pequeño y su grado de FGR), especialmente teniendo en cuenta, como se ha mencionado anteriormente que únicamente se utilizan 17 variables, de las cuales tan sólo 5 se obtuvieron de pruebas médicas de coste, el resto procedían de cuestionarios o fueron pruebas para medir la tensión, lo que hace mucho más sencilla la localización de datos para predicciones futuras, es la unión de los dos modelos del apartado 4.2.3.1.2.: *'xgboost_model1'* y *'xgboost_model2'*.

Asimismo, mencionar que únicamente se dispone de dos variables del ultrasonido del primer trimestre 'CRL' y 'BPD_First', y de estas dos únicamente se pudo utilizar 'CRL' debido a la gran cantidad de nulos de 'BPD_First', y 'CRL' fue descartada por no ser explicativa.

Sería muy aconsejable realizar y disponer de los datos del ultrasonido del primer trimestre para analizar si estas variables explican mejor el caso de estudio, lo que mejoraría la predicción.

Se puede comprobar que la precisión de este modelo final es de 80,35%, muy alta teniendo en cuenta que únicamente se cuenta con 17 variables, de las cuales tan sólo 5 se obtuvieron de pruebas médicas de coste, el resto procedían de cuestionarios o fueron pruebas para medir la tensión. Asimismo, mencionar que únicamente se dispone de dos variables del ultrasonido del primer trimestre 'CRL' y 'BPD_First', y de estas dos únicamente se pudo utilizar 'CRL' debido a la gran cantidad de nulos de 'BPD_First', y 'CRL' fue descartada por no ser explicativa.

```
#MODELO FINAL:
#Para cargar los datos únicamente se debe llamar al siguiente método:
y_pred1, model1 = cargar_validar_modelo('xgboost_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_model2', features1, y1)

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = data['FGR_birth_nivel']

#Para mostrar los reportes se llamará al siguiente método:
reportes_modelo_cargado(y_test, y_pred_final)
```

La precisión del modelo es: 0.8035413153456998.
[613 161 126 286]

Reporte de clasificación:				
	precision	recall	f1-score	support
0.0	0.91	0.94	0.93	613
1.0	0.62	0.66	0.64	161
2.0	0.52	0.79	0.62	126
3.0	0.90	0.60	0.72	286
accuracy			0.80	1186
macro avg	0.74	0.75	0.73	1186
weighted avg	0.83	0.80	0.80	1186



4.6.1.2. Fenotipos del modelo

Los fenotipos del modelo, para cada clase, son los siguientes:

Los fenotipos que explican que un neonato nazca pequeño (categoría 0) son: SBP_at_recruitment: 112.58 (desviación estándar: 10.05), Second_trim_Hto: 34.46 (desviación estándar: 2.57), Second_trim_Hb: 11.46 (desviación estándar: 0.76), DBP_at_recruitment: 72.02 (desviación estándar: 7.74), Maternal_pre_gestational_weight: 62.02 (desviación estándar: 10.11), GA_US_Second_trim: 20.8 (desviación estándar: 0.88), Placenta_pathology: 0 (% de representación: 74.71 %), EFW_SECOND: 394.0 (desviación estándar: 80.32), Maternal_age_at_the_time_of_recruitment: 32.61 (desviación estándar: 5.06), First_trim_Hto: 37.81 (desviación estándar: 2.74), AC_SECOND: 159.62 (desviación estándar: 13.51), PE: 0.0 (% de representación: 97.39 %), Smoke: 0.0 (% de representación: 68.52 %), Steroids_during_gestation: 0.0 (% de representación: 93.8 %), OSullivan_test_result: 122.41 (desviación estándar: 32.07), DBP_First_trim: 69.23 (desviación estándar: 8.72), SBP_First_trim: 110.51 (desviación estándar: 12.34)

Los fenotipos que explican que un neonato nazca con FGR Leve (categoría 1) son: SBP_at_recruitment: 114.2 (desviación estándar: 14.4), Second_trim_Hto: 34.88 (desviación estándar: 2.69), Second_trim_Hb: 11.61 (desviación estándar: 0.82), DBP_at_recruitment: 73.05 (desviación estándar: 10.32), Maternal_pre_gestational_weight: 59.94 (desviación estándar: 10.18), GA_US_Second_trim: 20.84 (desviación estándar: 1.17), Placenta_pathology: 0 (% de representación: 65.22 %), EFW_SECOND: 387.76 (desviación estándar: 105.52), Maternal_age_at_the_time_of_recruitment: 31.2 (desviación estándar: 6.05), First_trim_Hto: 38.19 (desviación estándar: 2.82), AC_SECOND: 158.23 (desviación estándar: 17.57), PE: 0.0 (% de representación: 87.58 %), Smoke: 0.0 (% de representación: 63.98 %), Steroids_during_gestation: 0.0 (% de representación: 95.03 %), OSullivan_test_result: 118.79 (desviación estándar: 33.8), DBP_First_trim: 70.31 (desviación estándar: 8.09), SBP_First_trim: 109.8 (desviación estándar: 10.66)

Los fenotipos que explican que un neonato nazca con FGR Leve (categoría 1) son: SBP_at_recruitment: 118.14 (desviación estándar: 14.27), Second_trim_Hto: 35.04 (desviación estándar: 2.71), Second_trim_Hb: 11.66 (desviación estándar: 0.86), DBP_at_recruitment: 74.19 (desviación estándar: 9.59), Maternal_pre_gestational_weight: 59.76 (desviación estándar: 11.4), GA_US_Second_trim: 20.81 (desviación estándar: 1.18), Placenta_pathology: 0 (% de representación: 53.17 %), EFW_SECOND: 384.21 (desviación estándar: 87.14), Maternal_age_at_the_time_of_recruitment: 31.35 (desviación estándar: 5.62), First_trim_Hto: 37.88 (desviación estándar: 2.59), AC_SECOND: 156.57 (desviación estándar: 14.44), PE: 0.0 (% de representación: 85.71 %), Smoke: 0.0 (% de representación: 68.25 %), Steroids_during_gestation: 0.0 (% de representación: 85.71 %), OSullivan_test_result: 119.68 (desviación estándar: 35.67), DBP_First_trim: 71.11 (desviación estándar: 8.78), SBP_First_trim: 112.51 (desviación estándar: 13.61)

Los fenotipos que explican que un neonato nazca con FGR Leve (categoría 1) son: SBP_at_recruitment: 127.2 (desviación estándar: 19.6), Second_trim_Hto: 35.87 (desviación estándar: 3.08), Second_trim_Hb: 11.88 (desviación estándar: 1.0), DBP_at_recruitment: 81.42 (desviación estándar: 12.86), Maternal_pre_gestational_weight: 60.35 (desviación estándar: 12.25), GA_US_Second_trim: 21.0 (desviación estándar: 1.23), Placenta_pathology: 1 (% de representación: 55.94 %), EFW_SECOND: 381.35 (desviación estándar: 85.77), Maternal_age_at_the_time_of_recruitment: 32.8 (desviación estándar: 5.76), First_trim_Hto: 38.1 (desviación estándar: 3.12), AC_SECOND: 156.69 (desviación estándar: 13.73), PE: 0.0 (% de representación: 61.89 %), Smoke: 0.0 (% de representación: 68.53 %), Steroids_during_gestation: 0.0 (% de representación: 64.69 %), OSullivan_test_result: 119.13 (desviación estándar: 35.17), DBP_First_trim: 73.94 (desviación estándar: 10.09), SBP_First_trim: 115.09 (desviación estándar: 12.51)

	Variables	No FGR	Std o %Repr 0	FGR 1	Std o %Repr 1	FGR 2	Std o %Repr 2	FGR 3	Std o %Repr 3
0	SBP_at_recruitment	112.58	10.05	114.20	14.4	118.14	14.27	127.20	19.6
1	Second_trim_Hto	34.46	2.57	34.88	2.69	35.04	2.71	35.87	3.08
2	Second_trim_Hb	11.46	0.76	11.61	0.82	11.66	0.86	11.88	1
3	DBP_at_recruitment	72.02	7.74	73.05	10.32	74.19	9.59	81.42	12.86
4	Maternal_pre_gestational_weight	62.02	10.11	59.94	10.18	59.76	11.4	60.35	12.25
5	GA_US_Second_trim	20.80	0.88	20.84	1.17	20.81	1.18	21.00	1.23
6	Placenta_pathology	0.00	74.71 %	0.00	65.22 %	0.00	53.17 %	1.00	55.94 %
7	EFW_SECOND	394.00	80.32	387.76	105.52	384.21	87.14	381.35	85.77
8	Maternal_age_at_the_time_of_recruitment	32.61	5.06	31.20	6.05	31.35	5.62	32.80	5.76
9	First_trim_Hto	37.81	2.74	38.19	2.82	37.88	2.59	38.10	3.12
10	AC_SECOND	159.62	13.51	158.23	17.57	156.57	14.44	156.69	13.73
11	PE	0.00	97.39 %	0.00	87.58 %	0.00	85.71 %	0.00	61.89 %
12	Smoke	0.00	68.52 %	0.00	63.98 %	0.00	68.25 %	0.00	68.53 %
13	Steroids_during_gestation	0.00	93.8 %	0.00	95.03 %	0.00	85.71 %	0.00	64.69 %
14	OSullivan_test_result	122.41	32.07	118.79	33.8	119.68	35.67	119.13	35.17
15	DBP_First_trim	69.23	8.72	70.31	8.09	71.11	8.78	73.94	10.09
16	SBP_First_trim	110.51	12.34	109.80	10.66	112.51	13.61	115.09	12.51

4.6.1.3. Significado de las variables seleccionadas

Las variables del modelo que dan respuesta al estudio son los siguientes:

```
variables_seleccionadas_estudio_FGR = 'SBP_at_recruitment',
'Second_trim_Hto','Second_trim_Hb','DBP_at_recruitment',
'Maternal_pre_gestational_weight','GA_US_Second_trim','Placenta_pathology','EFW_SECOND',
'Maternal_age_at_the_time_of_recruitment','First_trim_Hto','AC_SECOND','PE','Smoke','Steroids_during_gestation',
'OSullivan_test_result','DBP_First_trim','SBP_First_trim'
```

- **SBP_at_recruitment:** Presión arterial sistólica en la primera visita en el centro médico de estudio (esto no garantiza que tomaran la presión a la paciente en el primer trimestre de embarazo, sino en su primera visita al centro médico en que se le recogen los datos para la muestra).
- **Second_trim_Hto:** Hematocrito materno en el segundo trimestre.
- **Second_trim_Hb:** Hemoglobina materna en el segundo trimestre.
- **DBP_at_recruitment:** Presión arterial diastólica en el reclutamiento.
- **Maternal_pre_gestational_weight:** Peso de la madre antes del embarazo.
- **GA_US_Second_trim:** Edad gestacional en la ecografía del segundo trimestre en semanas.
- **Placenta_pathology:** Binaria que indica si existe alguna patología en la placenta (0 no, 1 sí).
- **EFW_SECOND:** Peso fetal estimado en el segundo trimestre calculado por ultrasonido en gramos.
- **Maternal_age_at_the_time_of_recruitment:** Edad materna en la primera visita en el centro médico de estudio.
- **First_trim_Hto:** Hematocrito materno en el primer trimestre.
- **AC_SECOND:** Circunferencia abdominal (del feto) en el segundo trimestre medida por ultrasonido en milímetros.

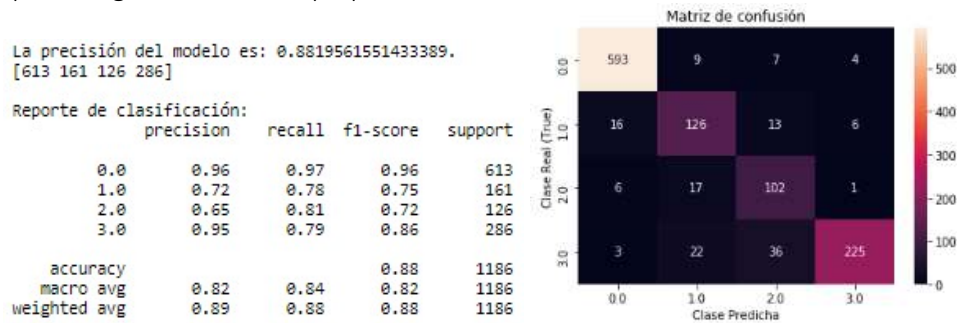
- **PE:** Diagnóstico de preeclampsia (0 no, 1 sí).
- **Smoke:** Binaria que toma valor 0 si la madre no fuma, y valor 1 si fuma.
- **Steroids_during_gestation:** Binaria que toma valor 0 si la madre no toma esteroides durante el embarazo y valor 1 si los toma.
- **OSullivan_test_result:** Resultado de la prueba de Osullivan que se utiliza para detectar la diabetes gestacional.
- **DBP_First_trim:** Presión arterial diastólica a las 11-13 semanas.
- **SBP_First_trim:** Presión arterial sistólica a las 11-13 semanas.

4.6.2. Modelo 2: Predicción no temprana de feto con problemas y grado de FGR con variables de la ecografía Doppler (Unión de 2 modelos)

4.6.2.1. Selección del modelo

El modelo seleccionado por sus buenos resultados de predicción (precisión de 88,19% prediciendo si un neonato nacerá pequeño y su grado de FGR), especialmente teniendo en cuenta que tan sólo se utilizan 13 variables, 6 nuevas que aportaría la ecografía Doppler y se mantienen 5 del modelo del apartado 4.2.3.1.2.

Dada la importancia de los fenotipos que aporta esta prueba al estudio, sería muy aconsejable realizar esta prueba a la mayor brevedad para disponer de los datos con la mayor antelación y contar con más tiempo para tratar de predecir cuanto antes si existe algún problema en el feto que le haga nacer antes o pequeño.



4.6.2.2. Fenotipos del modelo

	Variables	No FGR	Std o	%Repr 0	FGR 1	Std o	%Repr 1	FGR 2	Std o	%Repr 2	FGR 3	Std o	%Repr 3
0	IP_ACM_bX	1.90		0.34	1.87		0.36	1.81		0.36	1.74		0.46
1	PERCENTIL_EFW_bX	39.45		32.25	12.60		16.28	7.80		10.66	3.40		6.26
2	IP_mAut_bX	0.81		0.30	0.81		0.25	0.97		0.52	1.23		0.60
3	SBP_at_recruitment	112.58		10.05	114.20		14.40	118.14		14.27	127.20		19.60
4	Maternal_age_at_the_time_of_recruitment	32.61		5.06	31.20		6.05	31.35		5.62	32.80		5.76
5	OSullivan_test_result	122.41		32.07	118.79		33.80	119.68		35.67	119.13		35.17
6	Maternal_pre_gestational_weight	62.02		10.11	59.94		10.18	59.76		11.40	60.35		12.25
7	EFW_SECOND	394.00		80.32	387.76		105.52	384.21		87.14	381.35		85.77
8	First_trim_Hto	37.81		2.74	38.19		2.82	37.88		2.59	38.10		3.12
9	GA_US_Second_trim	20.80		0.88	20.84		1.17	20.81		1.18	21.00		1.23
10	IP_AU_bX	1.01		0.17	1.02		0.18	1.04		0.27	1.32		0.51
11	GA_US_Diagnosis_Fetal_Suboptimal_Growth	31.84		4.59	32.90		3.87	33.19		4.07	31.16		4.03
12	BPD_bX	77.90		11.38	78.43		8.53	78.45		8.69	72.67		10.74

4.6.2.3. Significado de las variables seleccionadas

Las variables del modelo, contenidas en **utilizar_periodico**, que dan respuesta al estudio son las siguientes:

utilizar_periodico = 'IP_ACM_DX', 'PERCENTIL_EFW_DX', 'IP_mAut_DX', 'SBP_at_recruitment', 'Maternal_age_at_the_time_of_recruitment', 'OSullivan_test_result', 'Maternal_pre_gestational_weight', 'EFW_SECOND', 'First_trim_Hto', 'GA_US_Second_trim', 'IP_AU_DX', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'BPD_DX'

- **IP_ACM_DX**: Medición del índice de pulsatilidad de la arteria cerebral media.
- **PERCENTIL_EFW_DX**: Percentil ajustado del peso fetal en gramos estimado en la ecografía.
- **IP_mAut_DX**: Medición del índice de pulsatilidad del Istmo aórtico.
- **SBP_at_recruitment**: Presión arterial sistólica en la primera visita en el centro médico de estudio.
- **Maternal_age_at_the_time_of_recruitment**: Edad materna en la primera visita en el centro médico de estudio.
- **OSullivan_test_result**: Resultado de la prueba de Osullivan que se utiliza para detectar la diabetes gestacional.
- **Maternal_pre_gestational_weight**: Peso de la madre antes del embarazo.
- **EFW_SECOND**: Peso fetal estimado en el segundo trimestre calculado por ultrasonido en gramos.
- **First_trim_Hto**: Hematocrito materno en el primer trimestre.
- **GA_US_Second_trim**: Edad gestacional en la ecografía del segundo trimestre en semanas.
- **IP_AU_DX**: Medición del índice de pulsatilidad de la arteria umbilical.
- **GA_US_Diagnosis_Fetal_Suboptimal_Growth**: Edad gestacional en gramos del feto en el ultrasonido.
- **BPD_DX**: Diámetro biparietal en milímetros medido por ultrasonido.

4.6.3. Modelo 2: Predicción no temprana de feto con problemas y grado de FGR con variables de la ecografía Doppler y ecografía fetal del tercer trimestre (Unión de 2 modelos)

4.6.3.1. Selección del modelo

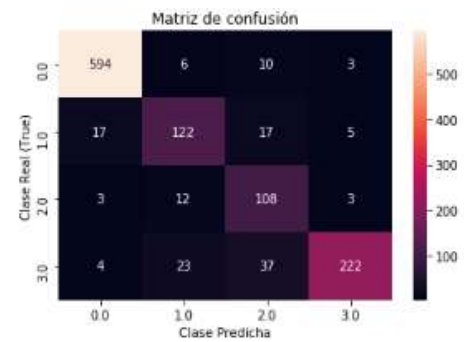
El modelo seleccionado por sus buenos resultados de predicción (precisión de 88,19% prediciendo si un neonato nacerá pequeño y su grado de FGR), especialmente teniendo en cuenta que tan sólo se utilizan 15 variables, 2 nuevas que aportaría la ecografía fetal del tercer trimestre y se mantienen las 13 del modelo del apartado **4.3.1.2**.

Como se puede comprobar, no se consiguió mejorar la precisión del modelo anterior, pero sí la del modelo agregando las variables de la ecografía del tercer trimestre, pero si se comparan las diferentes variables, se podrá ver que sólo se logró llegar al mismo resultado manteniendo las mismas variables que en el modelo del apartado **4.3.1.2**. y agregando dos nuevas variables de la ecografía del 3^{er} trimestre, las variables '**PERCENTIL_EFW_THIRDT**' y '**BPD_THIRDT**' (recordar que agregarlas todas empeoraba el modelo), lo cual hace que la selección de este modelo respecto del anterior (sección **4.3.1.2**.) sólo sea recomendable cuando no se disponga de todos o alguno de los datos de la ecografía Doppler que se mencionaba en el apartado anterior, ya que las variables explican tanta varianza que anulan la incidencia de las de esta prueba.

Dada la importancia de los fenotipos que aporta esta prueba al estudio, sería muy aconsejable realizar esta prueba a la mayor brevedad para disponer de los datos con la mayor antelación y contar con más tiempo para tratar de predecir cuanto antes si existe algún problema en el feto que le haga nacer antes o pequeño.

La precisión del modelo es: 0.8819561551433389.
[613 161 126 286]

Reporte de clasificación:				
	precision	recall	f1-score	support
0.0	0.96	0.97	0.97	613
1.0	0.75	0.76	0.75	161
2.0	0.63	0.86	0.72	126
3.0	0.95	0.78	0.86	286
accuracy			0.88	1186
macro avg	0.82	0.84	0.82	1186
weighted avg	0.89	0.88	0.88	1186



4.6.3.2. Fenotipos del modelo

	Variables	No FGR	Std o %Repr 0	FGR 1	Std o %Repr 1	FGR 2	Std o %Repr 2	FGR 3	Std o %Repr 3
0	PERCENTIL_EFW_DX	39.45	32.25	12.60	16.28	7.80	10.66	3.40	6.26
1	IP_ACM_DX	1.90	0.34	1.87	0.36	1.81	0.36	1.74	0.46
2	IP_mAut_DX	0.81	0.30	0.81	0.25	0.97	0.52	1.23	0.60
3	SBP_at_recruitment	112.58	10.05	114.20	14.40	118.14	14.27	127.20	19.60
4	First_trim_Hto	37.81	2.74	38.19	2.82	37.88	2.59	38.10	3.12
5	IP_AU_DX	1.01	0.17	1.02	0.18	1.04	0.27	1.32	0.51
6	GA_US_Diagnosis_Fetal_Suboptimal_Growth	31.84	4.59	32.90	3.87	33.19	4.07	31.16	4.03
7	Maternal_age_at_the_time_of_recruitment	32.61	5.06	31.20	6.05	31.35	5.62	32.80	5.76
8	OSullivan_test_result	122.41	32.07	118.79	33.80	119.68	35.67	119.13	35.17
9	GA_US_Second_trim	20.80	0.88	20.84	1.17	20.81	1.18	21.00	1.23
10	PERCENTIL_EFW_THIRDT	45.03	28.47	17.13	19.13	12.11	16.14	5.42	8.56
11	Maternal_pre_gestational_weight	62.02	10.11	59.94	10.18	59.76	11.40	60.35	12.25
12	BPD_THIRDT	82.84	4.74	81.66	4.82	81.55	4.57	78.54	6.09
13	EFW_SECONDT	394.00	80.32	387.76	105.52	384.21	87.14	381.35	85.77
14	BPD_DX	77.90	11.38	78.43	8.53	78.45	8.69	72.67	10.74

4.6.3.3. Significado de las variables seleccionadas

Los fenotipos del modelo, contenidos en 'utilizar_periodico_3trim', que dan respuesta al estudio son los siguientes:

utilizar_periodico_3trim = 'PERCENTIL_EFW_DX', 'IP_ACM_DX', 'IP_mAut_DX', 'SBP_at_recruitment', 'First_trim_Hto', 'IP_AU_DX', 'GA_US_Diagnosis_Fetal_Suboptimal_Growth', 'Maternal_age_at_the_time_of_recruitment', 'OSullivan_test_result', 'GA_US_Second_trim', 'PERCENTIL_EFW_THIRDT', 'Maternal_pre_gestational_weight', 'BPD_THIRDT', 'EFW_SECONDT', 'BPD_DX'

- **PERCENTIL_EFW_DX:** Percentil ajustado del peso fetal en gramos estimado en la ecografía.
- **IP_ACM_DX:** Medición del índice de pulsatilidad de la arteria cerebral media.
- **IP_mAut_DX:** Medición del índice de pulsatilidad del Istmo aórtico.
- **SBP_at_recruitment:** Presión arterial sistólica en la primera visita en el centro médico de estudio.
- **First_trim_Hto:** Hematocrito materno en el primer trimestre.
- **IP_AU_DX:** Medición del índice de pulsatilidad de la arteria umbilical.
- **GA_US_Diagnosis_Fetal_Suboptimal_Growth:** Edad gestacional en gramos del feto en el ultrasonido.
- **Maternal_age_at_the_time_of_recruitment:** Edad materna en la primera visita en el centro médico de estudio.
- **OSullivan_test_result:** Resultado de la prueba de Osullivan que se utiliza para detectar la diabetes gestacional.
- **GA_US_Second_trim:** Edad gestacional en la ecografía del segundo trimestre en semanas.
- **PERCENTIL_EFW_THIRDT:** Percentil ajustado del peso fetal estimado en gramos en el tercer trimestre.
- **Maternal_pre_gestational_weight:** Peso de la madre antes del embarazo.
- **BPD_THIRDT:** Diámetro biparietal en milímetros medido por ultrasonido.
- **EFW_SECONDT:** Peso fetal estimado en el segundo trimestre calculado por ultrasonido en gramos.
- **BPD_DX:** Diámetro biparietal en milímetros medido por ultrasonido.

5. Estudio adicional: Predicción del diagnóstico de los médicos sobre estimación de feto con problemas y grado de gravedad

Este apartado se dividirá en 2 fases:

1. **Predicción de juicio de los médicos:** Independientemente de que ya se haya seleccionado el mejor modelo que predice el nivel de FGR que tendrá un neonato (se han propuesto 3 dependiendo de los datos disponibles) se quiere predecir la variable 'Fetal_problem_1' para tratar de detectar qué variables hicieron a los médicos predecir ese diagnóstico para los neonatos. Para ello, se volverán a aplicar las técnicas de XGBoost, regresión logística y red neuronal.
2. **Comparación de aciertos entre juicio de los médicos y el modelo construido en este trabajo:** Para finalizar este apartado, se decide realizar un análisis sobre los aciertos de los médicos (variable 'Fetal_problem_before_delivery', que es equivalente a la variable 'Fetal_problem_1' binarizada, la cual no contiene nulos: 0 para previsión de neonato nacido sin problemas, y 1 para el resto) y la realidad (variable 'BW_Percentile_inferior_10').

Como se desconoce el porqué del diagnóstico realizado por los médicos, los cuales han realizado 8 clasificaciones diferentes, y para posibilitar la comparación entre realidad y predicción, se deberá binarizar la variable 'Fetal_problem_1', indicando 0 cuando los médicos hayan previsto que el neonato nacerá sin problemas, y 1 en el resto de casos. Esta información ya la contiene la variable 'Fetal_problem_before_delivery' y posibilitará comparar posteriormente la diferencia de acierto entre el juicio de los médicos y uno de los 3 modelos que se han construido.

No se realizará este estudio demasiado denso, ya que no forma parte del objetivo de este proyecto, únicamente se considera un agregado.

5.1. Predicción temprana de juicio de los médicos del nivel de gravedad del feto (variable Fetal_problem_1)

En el siguiente estudio se descartará del estudio la red neuronal, ya que se ha comprobado que no responde bien ante este problema, por lo que se utilizarán únicamente las técnicas de XGBoost y Regresión logística.

En el apartado 3. Estudio Predicción de los médicos (3 modelos) (Variable 'Fetal_problem_1') del anexo, se puede consultar otra manera de enfocar este estudio, uniendo 3 modelos para perder menos datos en la muestra al estratificar, en el que sí se incluye la red neuronal como análisis.


```
#'Fetal_problem_1'
print(data['Fetal_problem_1'].value_counts())

#Modelo 1: Predecir variable 'Fetal_problem_before_delivery', juicio de los médicos:
print('\n*****Modelo 1*****')

#Parámetros:
y = data['Fetal_problem_before_delivery']
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
y_column = 'Fetal_problem_before_delivery'

dataset1['Fetal_problem_before_delivery'] = dataset1['Fetal_problem_before_delivery'].astype('category')
print(dataset1['Fetal_problem_before_delivery'].value_counts())
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
xgboost_doctor_model1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'FGR_birth_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
la clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'Fetal_problem_1'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
#Renombrar las categorías:
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].replace({'Sin_complicaciones': 0,
    'Small_for_gestational_age': 1, 'Mild_congenital_anomaly': 2, 'FGR_stage_1': 3, 'FGR_stage_2': 4,
    'FGR_stage_3': 5, 'FGR_stage_4': 6})
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].astype('category')
y_test = dataset2['Fetal_problem_1']
dataset2 = dataset2[dataset2['Fetal_problem_1']!=0]
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
reg_logistica_doctor_model2 = get_regresion_logistica_model_individual_multiclass(features2, y2, y_column, muestra_estratificada)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
cargan:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_doctor_model1, 'xgboost_doctor_model1')
guardar_modelo(xgboost_doctor_model2, 'xgboost_doctor_model2')

y_pred1, model1 = cargar_validar_modelo('xgboost_doctor_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_doctor_model2', features1, y1) #Contiene todos los datos, features2 y y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = y_test

reportes_modelo_cargado(y_test, y_pred_final)

#'Fetal_problem_1'
print(data['Fetal_problem_1'].value_counts())

#Modelo 1: Predecir variable 'Fetal_problem_before_delivery', juicio de los médicos:
print('\n*****Modelo 1*****')

#Parámetros:
y = data['Fetal_problem_before_delivery']
dataset1 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
y_column = 'Fetal_problem_before_delivery'

dataset1['Fetal_problem_before_delivery'] = dataset1['Fetal_problem_before_delivery'].astype('category')
print(dataset1['Fetal_problem_before_delivery'].value_counts())
y = dataset1[y_column]
muestra_estratificada = False

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
reg_logistica_doctor_model1 = get_regresion_logistica_model_individual_multiclass(features1, y1, y_column, muestra_estratificada)

#Modelo 2: Predecir variable 'FGR_birth_nivel'. Se descarta la categoría 0, ya que está sobreajustando el modelo, y
esta clase ya se ha predicho en el Modelo 1, y únicamente se utilizará para entrenar el mismo número de observaciones de
la clase con menos observaciones de las categorías:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'Fetal_problem_1'
y = data[y_column]
dataset2 = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
#Renombrar las categorías:
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].replace({'Sin_complicaciones': 0,
    'Small_for_gestational_age': 1, 'Mild_congenital_anomaly': 2, 'FGR_stage_1': 3, 'FGR_stage_2': 4,
    'FGR_stage_3': 5, 'FGR_stage_4': 6})
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].astype('category')
y_test = dataset2['Fetal_problem_1']
dataset2 = dataset2[dataset2['Fetal_problem_1']!=0]
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].cat.remove_categories([0])
muestra_estratificada = True

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
reg_logistica_doctor_model2 = get_regresion_logistica_model_individual_multiclass(features2, y2, y_column, muestra_estratificada)

#MODELO FINAL:
#Se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
cargan:
print('\n*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(reg_logistica_doctor_model1, 'reg_logistica_doctor_model1')
guardar_modelo(reg_logistica_doctor_model2, 'reg_logistica_doctor_model2')

y_pred1, model1 = cargar_validar_modelo('reg_logistica_doctor_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('reg_logistica_doctor_model2', features1, y1) #Contiene todos los datos
5, features2 y y2 no contienen la categoría 0

y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = y_test

reportes_modelo_cargado(y_test, y_pred_final)
```

Respecto al modelo con XGBoost, se consigue obtener una precisión del 65,18% en la predicción temprana del juicio de los médicos, con las variables contenidas en **variables_seleccionadas_estudio_FGR**. Al disponer de tantas clasificaciones diferentes, no es posible construir la matriz de confusión correspondiente, pero se realizará el análisis de manera individual. Respecto al modelo 1, se consigue una precisión del 71,85% acertando 105 de 136 clasificaciones de la clase 0 y 66 de 102 de la clase 1. Respecto al modelo 2, se obtiene una precisión de tan sólo el 23,21%, pero al unir los dos modelos, la precisión aumenta al 65,18%.

Respecto al modelo de Regresión logística, se consigue obtener una precisión del 52,10% en la predicción temprana del juicio de los médicos, con las variables contenidas en **variables_seleccionadas_estudio_FGR**. Al disponer de tantas clasificaciones diferentes, no es posible construir la matriz de confusión correspondiente, pero se realizará el análisis de manera individual. Respecto al modelo 1, se consigue una precisión del 71,01% acertando 107 de 136 clasificaciones de la clase 0 y 62 de 102 de la clase 1. Respecto al modelo 2, se obtiene una precisión de tan sólo el 19,64%, pero al unir los dos modelos, la precisión aumenta al 52,10%.

5.2. Predicción temprana de juicio de los médicos de feto con problemas (variable Fetal_problem_before_delivery)

```
#'Fetal_problem_before_delivery'
#Modelo 2: Predecir variable 'Fetal_problem_before_delivery', es decir, si el neonato nace pequeño o no:
print('\n*****Modelo 2*****')

#Parámetros:
y_column = 'Fetal_problem_before_delivery'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False

features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
reg_logistica_model = get_regresion_logistica_model_individual_multiclass(features, y, y_column, muestra_estratificada)
```

```
#Fetal_problem_before_delivery'
#Modelo 2: Predecir variable 'Fetal_problem_before_delivery', juicio de los médicos:
print('*****Modelo 1*****')

#Parámetros:
y_column = 'Fetal_problem_before_delivery'
y = data[y_column]
dataset = pd.concat([variables_seleccionadas_estudio_FGR,y],axis=1)
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

print(dataset[y_column].value_counts()) #Se comprueba que la muestra está equilibrada
features, y = asignar_mediana_y_o_clase_aleatoria(dataset,y_column)
features = convertir_a_dummies(features)
xgboost_model = get_xgboost_model_individual(features, y, y_column, model_parameters)
```

Respecto al XGBoost, se consigue obtener una precisión de 74,02% en la predicción temprana del juicio de los médicos, con las variables de variables_seleccionadas_estudio_FGR. Se consigue acertar 104 de 136 clasificaciones de la clase 0 y 73 de 102 de la clase 1, lo cual es muy positivo para predicción temprana.

Respecto a la Regresión, se consigue obtener una precisión de 71,01% en la predicción temprana del juicio de los médicos, con variables_seleccionadas_estudio_FGR. Se consigue acertar 107 de 136 clasificaciones de la clase 0 y 62 de 102 de la clase 1, lo cual es muy positivo para una predicción temprana, pero se reduce la precisión en un aproximadamente un 3% respecto al mismo enfoque pero con la técnica XGBoost.

5.3. Predicción del grado de gravedad (predicción variable Fetal_problem_1)

5.3.1. Agregar variables perfiladas al modelo (Estudio 1)

```
#3: 'Fetal_problem_1'
#Modelo 1: Predecir variable 'Fetal_problem_before_delivery', juicio de los médicos:
print('*****Modelo 1*****')

#Parámetros:
y = data['Fetal_problem_before_delivery']
dataset1 = pd.concat([data[utilizar_periodico],y],axis=1)
y_column = 'Fetal_problem_before_delivery'

dataset1['Fetal_problem_before_delivery'] = dataset1['Fetal_problem_before_delivery'].astype('category')
print(dataset1['Fetal_problem_before_delivery'].value_counts())
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1,y_column)
features1 = convertir_a_dummies(features1)
xgboost_doctor_vars_periodico_model1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)

#Modelo 2: Predecir variable 'Fetal_problem_1'. Se descartará la categoría 0, ya que está sobreajustando el mo
#ya se ha predicho en el Modelo 1, y únicamente se utilizarán para entrenar el mismo número de observaciones d
#menos observaciones de las categorías:
print('*****Modelo 2*****')

#Parámetros:
y_column = 'Fetal_problem_1'
y = data[y_column]
dataset2 = pd.concat([data[utilizar_periodico],y],axis=1)

#Renombrar las categorías:
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].replace({'sin_complicaciones': 0,
    'Small_for_gestational_age': 1, 'Mild_congenital_anomaly': 2, 'FGR_stage_1': 3, 'FGR_stage_2': 4,
    'FGR_stage_3': 5, 'FGR_stage_4': 6})
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].astype('category')
y_test = dataset2['Fetal_problem_1']
dataset2 = dataset2[dataset2['Fetal_problem_1']!=0]
dataset2['Fetal_problem_1'] = dataset2['Fetal_problem_1'].cat.remove_categories([0])
muestra_estratificada = True
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}

features2, y2 = asignar_mediana_y_o_clase_aleatoria(dataset2,y_column)
features2 = convertir_a_dummies(features2)
xgboost_doctor_vars_periodico_model2 = get_xgboost_model_individual(features2, y2, y_column, model_parameters)

#MODELO FINAL:
#se une el resultado de los dos modelos en uno único. Para ello, primero se guardan los modelos, y después se
print('*****Modelo final - Unión del modelo 1 y modelo 2*****')
guardar_modelo(xgboost_doctor_vars_periodico_model1, 'xgboost_doctor_vars_periodico_model1')
guardar_modelo(xgboost_doctor_vars_periodico_model2, 'xgboost_doctor_vars_periodico_model2')

y_pred1, model1 = cargar_validar_modelo('xgboost_doctor_vars_periodico_model1', features1, y1)
y_pred2, model2 = cargar_validar_modelo('xgboost_doctor_vars_periodico_model2', features1, y1) #Contiene todos
y_pred_final = get_y_pred_modelo_final(y_pred1, y_pred2)
y_test = y_test
reportes_modelo_cargado(y_test, y_pred_final)
```

La incorporación de las nuevas variables de la ecografía Doppler mejora los resultados de predicción del modelo. Se consigue obtener una precisión de 69,22% respecto al 65,18% obtenido en la predicción temprana del juicio de los médicos. Respecto al modelo 1, existe una mejora notable, se pasa de un 71,85% de precisión a un 84,87%, pasando de acertar 105 de 136 clasificaciones de la clase 0 y 66 de 102 de la clase 1 a 115 de 136 en la clase 0 y 87 de 102 en la clase 1. Respecto al modelo 2, se obtiene una precisión de tan sólo el 25,89%, pero al unir los dos modelos, la precisión aumenta al 69,22%. Efectivamente, los valores de AUC y de gini del modelo 1 son muy positivos.

5.3.2. Agregar variables perfiladas al modelo (Estudio 2)

El código es el mismo que el del apartado anterior, sólo cambian los datos de entrada. La incorporación de las nuevas variables de la ecografía fetal del tercer trimestre mejora los resultados de predicción del modelo. Se consigue obtener una pequeña mejora de un 1,4% en la precisión respecto a la del modelo anterior. Concretamente, la precisión obtenida es de 70,66%. Respecto al modelo 1, existe una mejora notable en la precisión, pasando a acertar 122 de 136 clasificaciones de la clase 0 y 92 de 102 de la clase 1, aumentando así la precisión del modelo a un 89,92%. Respecto al modelo 2, se obtiene una precisión de tan sólo 34,82%, pero al unir los dos modelos, la precisión aumenta al 81,13%, lo que lo convierte en un muy buen modelo que consigue clasificar hasta 7 clases diferentes, clase 0 para predicción de neonato nacido en un percentil normal, 4 clasificaciones diferentes de restricción y 2 de neonato nacido con problemas pero sin restricción de crecimiento. Efectivamente, los valores de AUC y de gini del modelo son muy positivos.

5.4. Predicción de juicio de los médicos de neonato nacido con problemas (variable Fetal_problem_before_delivery)

5.4.1. Agregar variables perfiladas al modelo (Estudio 1)

```
#3. 'Fetal_problem_before_delivery'
#Modelo 1: Predecir variable 'Fetal_problem_before_delivery', juicio de los médicos:
print('*****Modelo 1*****')
#Parámetros:
y = data['Fetal_problem_before_delivery']
dataset1 = pd.concat([data[utilizar_periodico],y],axis=1)
y_column = 'Fetal_problem_before_delivery'

dataset1['Fetal_problem_before_delivery'] = dataset1['Fetal_problem_before_delivery'].astype('category')
print(dataset1['Fetal_problem_before_delivery'].value_counts())
y = dataset1[y_column]
muestra_estratificada = False
model_parameters = best_parameters_XGBoost_model1
model_parameters = {
    'max_depth': model_parameters['max_depth'],
    'n_estimators': model_parameters['n_estimators'],
    'learning_rate': model_parameters['learning_rate']
}
features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1, y_column)
features1 = convertir_a_dummies(features1)
xgboost_model1 = get_xgboost_model_individual(features1, y1, y_column, model_parameters)
```

La incorporación de las nuevas variables de la ecografía Doppler mejora los resultados de predicción del modelo respecto al obtenido en el apartado de detección temprana. Se consigue obtener una precisión de 84,87% respecto a la de 74,02% del modelo anterior. Se consiguen acertar 115 de 136 clasificaciones de la clase 0 y 87 de 102 de la clase 1. Se tienen unos valores altos de AUC y de gini.

5.4.2. Agregar variables perfiladas al modelo (Estudio 2)

El código es el mismo que el del apartado anterior, sólo cambian los datos de entrada. La incorporación de las nuevas variables de la ecografía Doppler mejora los resultados de predicción del modelo respecto al obtenido en el apartado 10.4.1.1. de detección temprana. Se consigue obtener una precisión de 89,92% respecto a la de 84,87% del modelo anterior, mejorando la precisión en aproximadamente un 5%. Se consiguen acertar 122 de 136 clasificaciones de la clase 0 y 92 de 102 de la clase 1. Se obtienen unos valores altos de AUC y de gini.

5.5. Estudio comparativo de aciertos entre juicio de los médicos y el modelo construido en este trabajo

En este estudio se realizará una comparación entre el modelo que se ha construido en el apartado 9.1.2.2., unión de los modelos xgboost_model_vars_periodico_3trim1_2 y xgboost_model_vars_periodico_3trim2_2, y la predicción realizada por los médicos, contenida en la variable 'Fetal_problem_before_delivery'. No se puede realizar la comparación con el grado de restricción, ya que las categorías en cada clase objetivo son diferentes. Hay que recordar que los 3 modelos seleccionados se han construido bajo la variable 'FGR_birth_nivel', que contiene 4 categorías diferentes, y la variable 'Fetal_problem_1' con la predicción de los médicos contiene 7 categorías diferentes, por lo que la única comparación posible es con las variables binarizadas, es decir, las variables 'BW_Percentile_inferior_10' y 'Fetal_problem_before_delivery'.

5.5.1. Modelo construido en este estudio (Apartado 4.6.3)

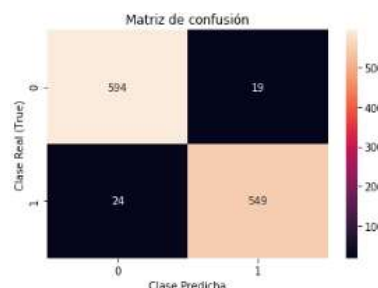
```
#Preparar los datos de entrada:
y_column = 'BW_Percentile_inferior_10'
y = data[y_column]
dataset1 = pd.concat([data[utilizar_periodico_3trim],y],axis=1)
features1, y1 = asignar_mediana_y_o_clase_aleatoria(dataset1, y_column)
features1 = convertir_a_dummies(features1)

y_test = y
y_pred_modelo, model1 = cargar_validar_modelo('xgboost_model_vars_periodico_3trim1_2', features1, y1)

print(np.unique(y_pred_modelo))
print(np.unique(y_test))
reportes_modelo_cargado(y_test, y_pred_modelo)
```

La precisión del modelo (Accuracy) es: 96.3743676222597
[613 573]

Reporte de clasificación:	precision	recall	f1-score	support
0	0.96	0.97	0.97	613
1	0.97	0.96	0.96	573
accuracy			0.96	1186
macro avg	0.96	0.96	0.96	1186
weighted avg	0.96	0.96	0.96	1186



Con el modelo construido, se obtiene una precisión de 96,37%, clasificando correctamente 594 de 613 clases de categoría 0 y 549 de 573 de clase 1, lo que ofrece unos resultados muy positivos en la predicción, ya que únicamente se fallan 19 clasificaciones de 613 para la clase 0 y 24 de 573 para la clase 1.

5.5.2. Diagnóstico de los médicos (Variable Fetal_problem_before_delivery)

```
##Médicos:
y_test = y
y_pred_medicos = data['Fetal_problem_before_delivery']
reportes_modelo_cargado(y_test, y_pred_medicos)
```

La precisión del modelo (Accuracy) es: 82.54637436762225
[613 573]

Reporte de clasificación:				
	precision	recall	f1-score	support
0	0.82	0.84	0.83	613
1	0.83	0.81	0.82	573
accuracy			0.83	1186
macro avg	0.83	0.82	0.83	1186
weighted avg	0.83	0.83	0.83	1186



Con los mismos datos utilizados en el apartado anterior, el juicio de los médicos obtiene una precisión de 82,55%, clasificando correctamente 516 de 613 clases de categoría 0 y 463 de 573 de clase 1, fallando 97 clasificaciones de la clase 0 y 110 de la clase 1.

Como conclusión del estudio, se determina que el modelo construido en este proyecto es mucho más preciso que el propio juicio de los médicos, para predecir si un neonato nacerá pequeño y/o con restricción.

6. Líneas de trabajo futuras

Después de desarrollar el proyecto, se considera como posibles estudios futuros los siguientes puntos:

- Estudio más profundo de la prematuridad
- Utilizar los datos del set de datos de entrega que correspondían a datos posteriores al parto, con el fin de analizar la evolución de los neonatos según su nivel de restricción o bien si nacieron sanos.
- Estudio de los distintos tipos de restricción del crecimiento, según su forma [10]. Este análisis se iba a desarrollar en este proyecto, pero finalmente no fue posible debido a que se dio prioridad a otros estudios.

La idea era crear nuevas variables a partir de las variables existentes, concretamente a partir de las variables “HC” y “AC”, que corresponden a la circunferencia de la cabeza del feto en milímetros y a la circunferencia abdominal del feto en milímetros. Según la bibliografía consultada, el FGR simétrico es aquel que se da cuando el feto presenta una proporción normal

entre el tamaño de la cabeza del feto y su cuerpo. FGR Simétrico suele generarse por tener un menor número de células en el cuerpo. El FGR es asimétrico cuando el feto presenta una desproporción entre el tamaño de la cabeza del feto y su cuerpo, siendo el tamaño de la cabeza muy superior al del cuerpo. Este tipo de FGR suele generarse por insuficiencia placentaria.

El desarrollo que se había planteado era consultar mediante las tablas de crecimiento los valores normales para esas dos variables, y consultar la semana de gestación equivalente según esos valores. El siguiente paso era comparar si el número de semanas de gestación eran equivalentes entre esas dos variables o había diferencias entre ellas y, en ese caso, si la semana de gestación de la variable "HC" era superior al de "AC" para identificar un FGR asimétrico.

Asimismo, sería muy interesante disponer de los datos de las ecografías del primer trimestre para analizar el impacto en los resultados de los modelos construidos.

7. Conclusiones

Mediante el desarrollo de este proyecto, se han logrado cumplir los objetivos iniciales del mismo. Se han conseguido encontrar los fenotipos que predicen con una precisión del 80,35%, y de manera precoz, el nacimiento de un neonato pequeño, es decir, con un percentil de crecimiento inferior a 10, además de predecir el grado de restricción con el que nacerá, FGR 1 o Leve (entre 5 y 10), FGR 2 Moderado (entre 2 y 5) o FGR 3 o Severo (inferior a 2), mediante la construcción de un modelo con la técnica XGBoost.

Asimismo, se han conseguido construir dos modelos adicionales que mejoran la precisión en la predicción, logrando predecir lo mismo que en primero modelo, pero mejorando la precisión en un 7,85%, consiguiendo así una precisión del 88,20%, con el coste de depender de una o dos pruebas médicas, que corresponden a ecografías realizadas en el tercer trimestre, y que por tanto retrasan la predicción. A este estudio se le llamó "predicción no temprana", y el número de días de media entre la fecha de obtención de estas pruebas y la fecha de parto fue entre 42 y 34 días.

Además, se ha realizado un estudio para predecir el juicio de los médicos, y qué variables eran influyentes. Se han logrado encontrar estas variables y posteriormente, se ha realizado un estudio comparativo para comprobar si acertaba más uno de los modelos construidos en el proyecto, que el juicio de los médicos con las mismas variables de entrada, y el resultado ha sido muy grato, ya que el modelo construido es un 13,82% más preciso que el diagnóstico de los médicos, logrando un 96,37% de precisión frente al 82,56% de los médicos.

Después del estudio se llega a la conclusión que sería muy beneficioso disponer de las ecografías a la mayor brevedad, una posible opción sería adelantar las pruebas del tercer trimestre con el fin de disponer de los datos con prematuridad y adelantar así las predicciones.

Como aportación personal, ha sido muy satisfactorio realizar este proyecto y vivir tan de cerca un tema tan profundo, delicado y emotivo como es el de los neonatos que nacen pequeños, tratando de encontrar cualquier factor, que de alguna manera pueda ayudar a predecir cuándo un neonato va a nacer con un percentil bajo, con el fin de intentar evitar por todos los medios que ese hecho ocurra, para que ese neonato y su familia tengan una vida mejor.

8. Glosario

- Angiogénesis: Es el proceso fisiológico que consiste en la formación de vasos sanguíneos nuevos a partir de los vasos preexistentes.
- Cromosoma: Se denomina a cada una de las estructuras altamente organizadas, formadas por ADN y proteínas, que contiene la mayor parte de la información genética de un ser vivo.
- Data mining: Conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.
- Deep learning: Conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial.
- Diabetes gestacional: La diabetes gestacional se manifiesta durante el embarazo (gestación). Al igual que con otros tipos de diabetes, la diabetes gestacional afecta la forma en que las células utilizan el azúcar (glucosa). La diabetes gestacional causa un alto nivel de azúcar en sangre que puede afectar tu embarazo y la salud del neonato.
- Ecografía Doppler: Prueba no invasiva que calcula el flujo de la sangre en los vasos sanguíneos haciendo rebotar ondas sonoras de alta frecuencia (ecografía) en los glóbulos rojos circulantes. En la ecografía común, se utilizan ondas sonoras para crear imágenes, pero no se puede mostrar el flujo sanguíneo.
- Espina bífida: Es cuando la columna no cierra adecuadamente, y el neonato no podrá caminar de forma normal.
- Fenotipo: Manifestación externa de un conjunto de caracteres visibles que un individuo presenta como resultado de la interacción entre su genotipo y el medio.
- FGR (Intrauterine Growth Restriction): Se da cuando un neonato que todavía está dentro del vientre materno (feto) no consigue crecer al ritmo esperado durante el embarazo.
- Hipoxia: Estado de deficiencia de oxígeno en la sangre, células y tejidos del organismo, que pueden comprometer su funcionamiento.
- Hormona Gonadotropina coriónica: Hormona que normalmente se produce en el cuerpo durante el embarazo.
- Inserción velamentosa del cordón: es una complicación del cordón umbilical en la que ésta llega a insertarse en la placenta a través de la superficie de las membranas ovulares.
- Líquido amniótico: es un fluido líquido que rodea y amortigua al embrión y luego al feto en desarrollo en el interior del saco amniótico. Permite al feto moverse dentro de la pared del útero sin que las paredes de este se ajusten demasiado a su cuerpo, además de proporcionarle sustentación hidráulica.

- Macrosomía: Término que se utiliza de forma imprecisa para definir a un feto que es grande.
- Morbilidad: Cantidad de personas que enferman en un lugar y un período de tiempo determinados en relación con el total de la población.
- Micro ARN (miRNA): Es un ARN monocatenario, de una longitud de entre 21 y 25 nucleótidos, y que tiene la capacidad de regular la expresión de otros genes mediante diversos procesos, utilizando para ello la ruta de ribointerferencia.
- Percentil: En medicina se utiliza el percentil como medida de distribución para poder tomar la población normal, distribuida y entonces analizar aproximadamente dentro de esta distribución al feto que se está tomando de ejemplo, según su peso y estatura. Normalmente, la mayoría de la población se encuentra en el percentil 50.
- Perinatal: Hechos o fenómenos ocurridos alrededor del nacimiento, bien sea antes, durante o después del mismo los cuales afectan o conciernen al neonato.
- Preeclampsia: Es la presión arterial alta y signos de daño hepático o renal que ocurren en las mujeres después de la semana 20 de embarazo. En menor frecuencia, la preeclampsia también se puede presentar en una mujer después de dar a luz a su neonato, casi siempre dentro de las siguientes 48 horas. Esto se denomina preeclampsia posparto.
- Prematuridad: Se considera prematuro a un neonato que ha nacido antes de que se cumpla la semana número 37 de gestación.
- Proteína plasmática unida al embarazo: Proteína circulante en el suero de mujeres con gestaciones avanzadas.
- SGA (Small for Gestational Age): Pequeño para la edad gestacional, se considera a un neonato que en el momento de su nacimiento es pequeño de peso y de altura. Esto se considera así cuando el neonato ha crecido por debajo del percentil 10.
- Síndrome antifosfolípido o síndrome de anticuerpos antifosfolípidos: Cuadro autoinmunitario de hipercoagulabilidad causado por anticuerpos dirigidos contra proteínas de unión a los fosfolípidos de las membranas celulares.
- Teratógeno: Agente capaz de causar un defecto congénito. Generalmente, se trata de algo que es parte del ambiente al que está expuesta la madre durante el embarazo. Puede ser un medicamento, una droga ilícita, el consumo de alcohol o una enfermedad de la madre capaz de aumentar la probabilidad de que el neonato nazca con un defecto congénito.
- Tubo neural: Estructura presente en el embrión, del que se origina el sistema nervioso central. Todos los embriones de vertebrados tienen un tubo neural antes de que se desarrolle su sistema nervioso central, y es básicamente el "primer borrador" del cerebro y la médula espinal.

9. Bibliografía

- [1] (KidsHealth, 2020) Restricción del crecimiento intrauterino <https://kidshealth.org/MainLine/es/parents/FGR-esp.html>
Fecha visita: 25/02/2020
- [2] (La Opinión ,2018) Uno de cada 13 bebés es prematuro en España <https://www.laopiniondemurcia.es/vida-y-estilo/salud/2018/07/03/13-bebes-prematuro-espana/935479.html>
Fecha visita: 23/02/2020
- [3] (FEAD, 2020) En España nacen 28.000 bebés prematuros al año <https://www.saludigestivo.es/en-espana-nacen-28-000-bebes-prematuros-al-ano-una-de-las-tasas-mas-altas-de-la-union-europea/>
Fecha visita: 26/02/2020
- [4] (Wikipedia) Cross Industry Standard Process for Data Mining https://es.m.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining
Fecha visita: 22/02/2020
- [5] (González y Arroyas ,2016) Prematuros: Problemas más frecuentes <https://enfamilia.aeped.es/edades-etapas/prematuros-problemas-mas-frecuentes>
Fecha visita: 21/02/2020
- [6] (Egon, 2020) 10 técnicas y ejemplos prácticos de data mining <https://www.egon.com/es/blog/667-tecnicas-data-mining-marketing>
Fecha visita: 22/02/2020
- [7] (Sinexus) Datamining. https://sinexus.com/business_intelligence/datamining.aspx
Fecha visita: 23/02/2020
- [8] (Álvarez, 2019) Problemas de crecimiento fetal, FGR. <https://www.youtube.com/watch?v=eYPfk9S-TY4>
Fecha visita: 15/03/2020
- [9] (Monsalve, 2019) Consejos de cuidado en el primer trimestre de embarazo. <https://www.youtube.com/watch?v=9WUYMboyKAE>
Fecha visita: 17/03/2020
- [10] (Moquillaza, 2018) FGR, tipos, causas y efectos. <https://www.youtube.com/watch?v=CbK4lvooHxE>
Fecha visita: 20/03/2020
- [11] (Serna, 2017) Causas y efectos de nacimiento de bebés de bajo peso (FGR y prematuridad). <https://www.youtube.com/watch?v=opdNffvySzM>
Fecha visita: 20/03/2020
- [12] (Cruz, 2019) FGR, etapas, causas, efectos y tratamientos para FGR. <https://www.youtube.com/watch?v=RVyu-JzElzo>
Fecha visita: 16/03/2020

- [13] (García, 2019) FGR, causas, clasificación, diagnóstico y efectos. <https://www.youtube.com/watch?v=ZaD9jzIwkjQ>
Fecha visita: 26/03/2020
- [14] (Osvaldo, 2016) Amenaza de prematuridad, definición, causas, efectos y tratamiento. <https://www.youtube.com/watch?v=43pCpZ-lo5o>
Fecha visita: 20/03/2020
- [15] (Bührer, 2016) Prematuridad, causas y factores. <https://www.youtube.com/watch?v=tyMHQTxOrB4>
Fecha visita: 16/03/2020
- [16] (Hirsch, 2014) FGR, tipos, causas y efectos. <https://kidshealth.org/en/parents/FGR.html>
Fecha visita: 19/03/2020
- [17] (Asociación Española de Pediatría, 2016) Prematuridad, problemas más frecuentes. <https://enfamilia.aeped.es/edades-etapas/prematurados-problemas-mas-frecuentes>
Fecha visita: 21/03/2020
- [18] (Vygon, 2018) Evolución prematuridad en España. https://blog.vygon.es/dia-mundial-prematurados/?fbclid=IwAR1XU70XZB_VL1j9maSEUtJdy21AfyKiBSCcD-Fiipf4KHJgoLk71SeISG4
Fecha visita: 15/03/2020
- [19] (INE, 2019) Prematuridad 2019, relación entre edad de la madre y prematuridad. <https://www.ine.es/jaxi/Datos.htm?path=/t20/e301/nacim/a2018/l0/&file=01011.px#!tabs-tabla>
Fecha visita: 15/03/2020
- [20] (Croatian Med J., 2013) Efecto del ejercicio materno durante el embarazo sobre el crecimiento fetal anormal. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3760660/>
Fecha visita: 19/03/2020
- [21] (Bentley, Shneuer, Lain, Martin, Gordon y Nassar, 2018) Estudio de la morbilidad neonatal prematura. <https://pediatrics.aappublications.org/content/141/2/e20171726>
Fecha visita: 21/03/2020
- [22] (Reproductive Health, 2018) Prematuridad, ¿somos conscientes de qué es? <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6136169/>
Fecha visita: 25/03/2020
- [23] (Crispi, Bijnens, Figueras, Bartrons, Eixarch, Le Noble, Ahmed y Gratacós, 2010) FGC se traduce en corazones remodelados y menos eficientes en niños. <https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.110.937995>
Fecha visita: 31/03/2020
- [24] (Crispi, Paules, Noell, Youssef, Crovetto, Simoes, Eixarch, Faner y Gratacós, 2019) identificación de fenotipos de restricción del crecimiento fetal utilizando un enfoque de agrupamiento basado en la red <https://obgyn.onlinelibrary.wiley.com/doi/full/10.1002/uog.20524>
Fecha visita: 31/03/2020

[25] (Chiu, Mitra, Boymoushakian y Collier, 2018) Ejemplo de aplicación del método SNF en detección de tumores en pacientes. https://www.researchgate.net/figure/Similarity-network-fusion-SNF-based-integrative-clustering-of-patient-tumors-A_fig2_326879642

Fecha visita: 31/03/2020

[26] (Wang, Mezlini y Demir, 2014) Método SNF <https://www.nature.com/articles/nmeth.2810>

Fecha visita: 31/03/2020

[27] (Fonteiijn, Modat, Clarkson, Barnes, Lehmann, Hobbs, Scahill, Tabrizi, Ourselin, Fox y Alexander, 2012) Un modelo basado en eventos para la progresión de la enfermedad y su aplicación en la enfermedad de Alzheimer familiar y la enfermedad de Huntington. <https://www.ncbi.nlm.nih.gov/pubmed/22281676>

Fecha visita: 31/03/2020

[28] (Young, Cash, Benzinger, Fagan, Morris, Bateman, Fox, Schott y Alexander, 2018) Modelos basados en datos de progresión de la enfermedad de Alzheimer predominantemente heredada.

<https://academic.oup.com/brain/article/141/5/1529/4951528>

Fecha visita: 31/03/2020

[29] (Young, Oxtoby, Daga, Cash, Fox, Ourselin, Schott y Alexander, 2014) Un modelo basado en datos de cambios de biomarcadores en la enfermedad de Alzheimer esporádica <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4132648/>

Fecha visita: 31/03/2020

[30] (Venkatraghavan, Dubost, Bron, Niessen, de Bruijne y Klein, 2019) Event-based-models (EBM). <https://arxiv.org/abs/1903.03386>

Fecha visita: 31/03/2020

[31] (Villena, 2016) CRISP-DM: La metodología para poner orden en los proyectos.

<https://www.sngular.com/es/data-science-crisp-dm-metodologia/>

Fecha visita: 11/03/2020

[32] (Raona Enginyers, 2017) Los 10 Algoritmos esenciales en Machine Learning.

<https://www.talend.com/resources/data-mining-techniques/>

Fecha visita: 12/03/2020

[33] (Talend) 16 técnicas de Data Mining. <https://www.raona.com/los-10-algoritmos-esenciales-machine-learning/>

Fecha visita: 12/03/2020

[34] (Palaggi, 2018) Jira y Confluence en proyectos. <https://medium.com/@brianpalaggi/why-does-your-team-needs-jira-and-confluence-26ffc8dde86b>

Fecha visita: 10/03/2020